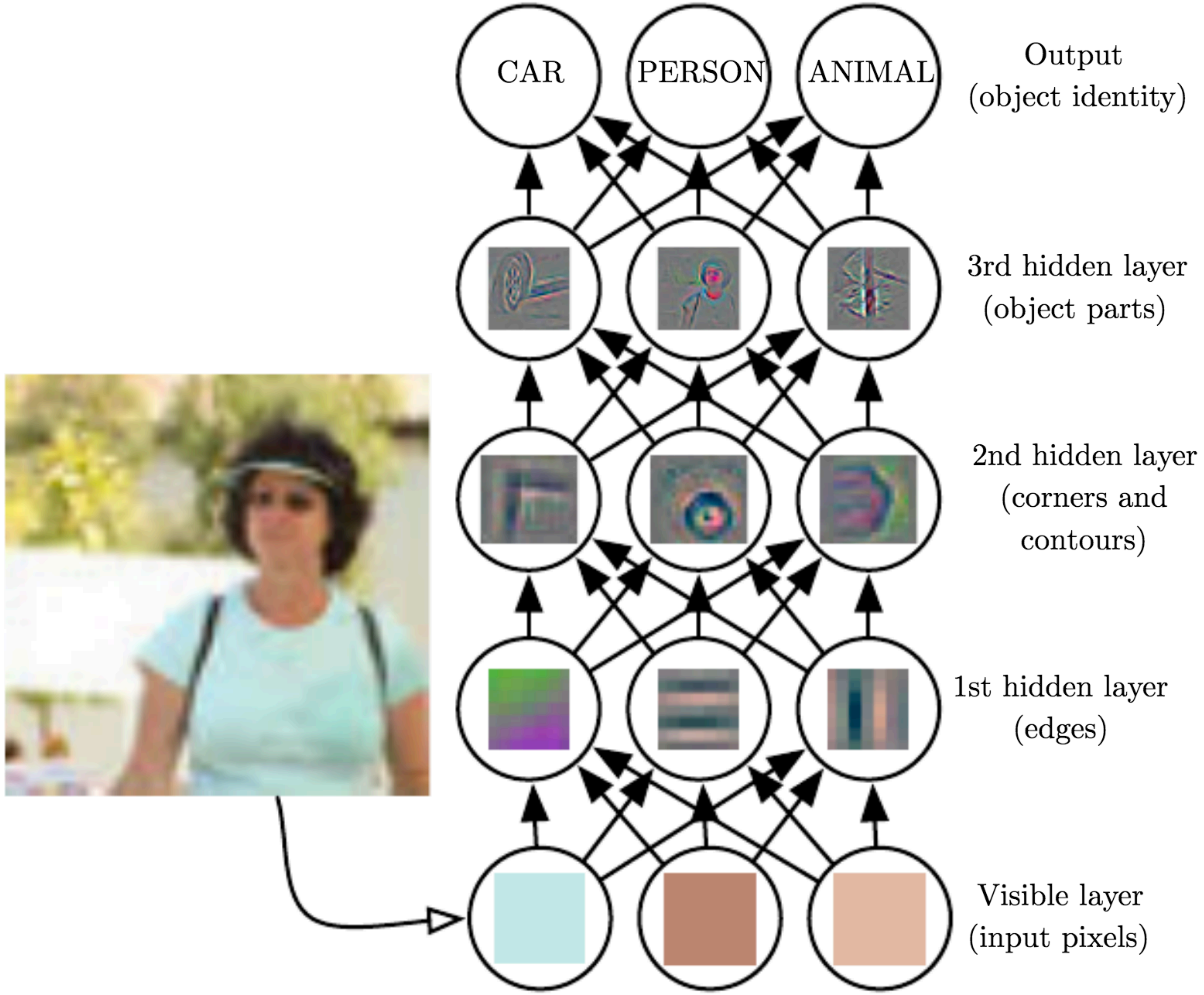
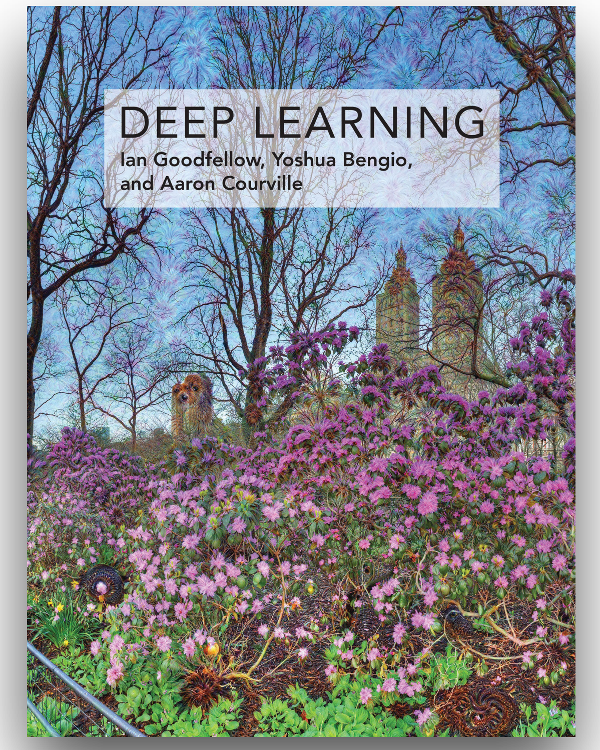
Neural Network Renormalization Group

Lei Wang (王磊)
Institute of Physics, CAS
https://wangleiphy.github.io

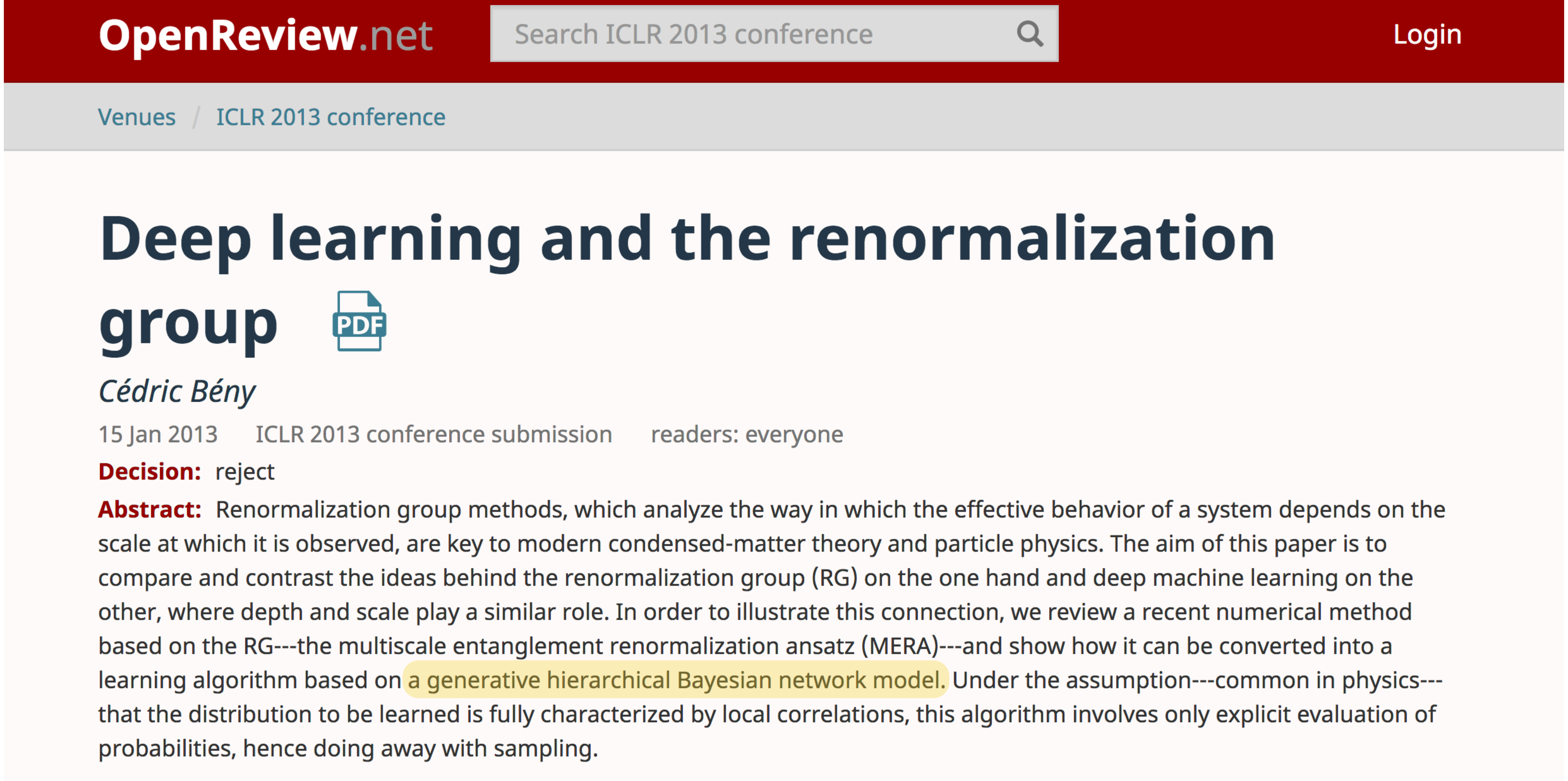
RG and Deep Learning



Page 6
Figure 1.2



Goodfellow, Bengio, Courville, http://www.deeplearningbook.org/



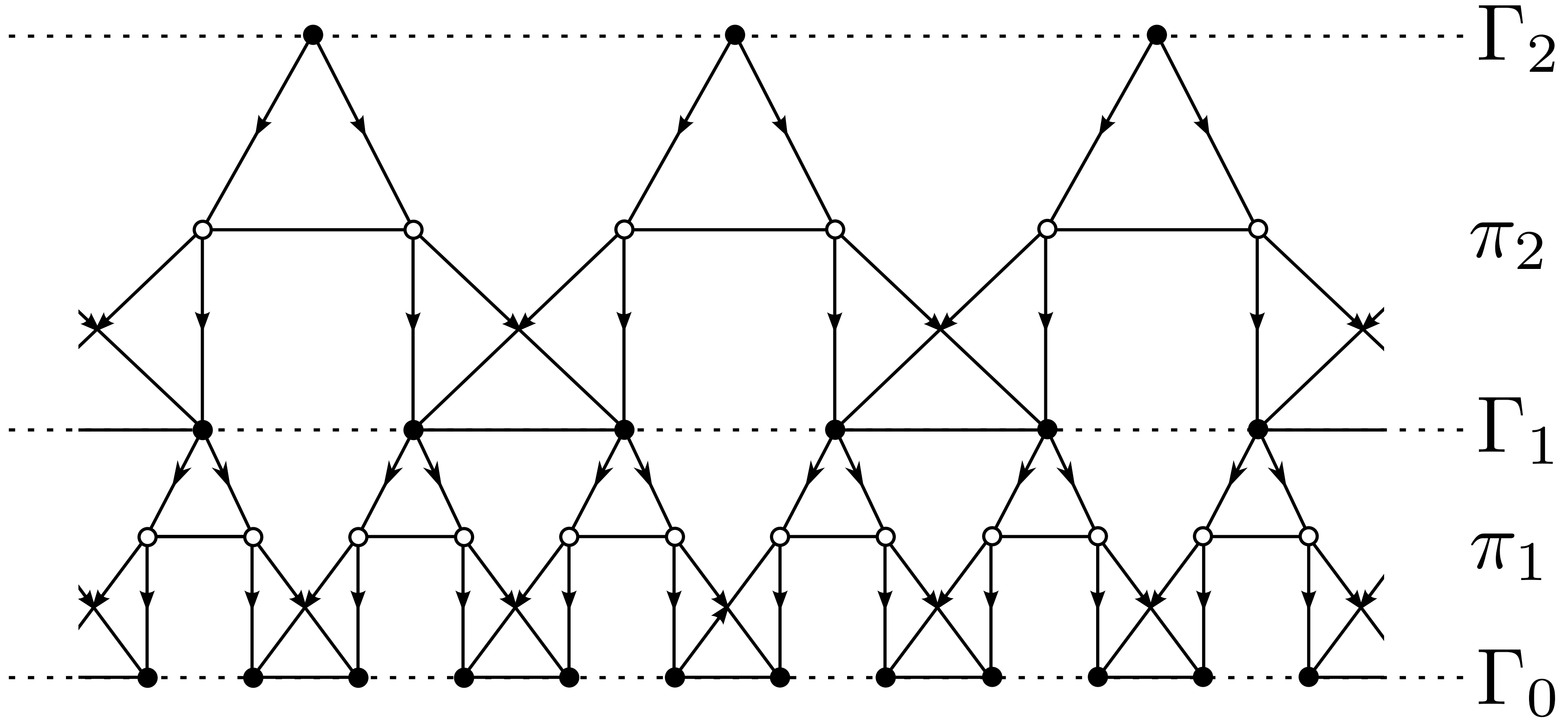
Deep learning and the renormalization group

Cédric Bény

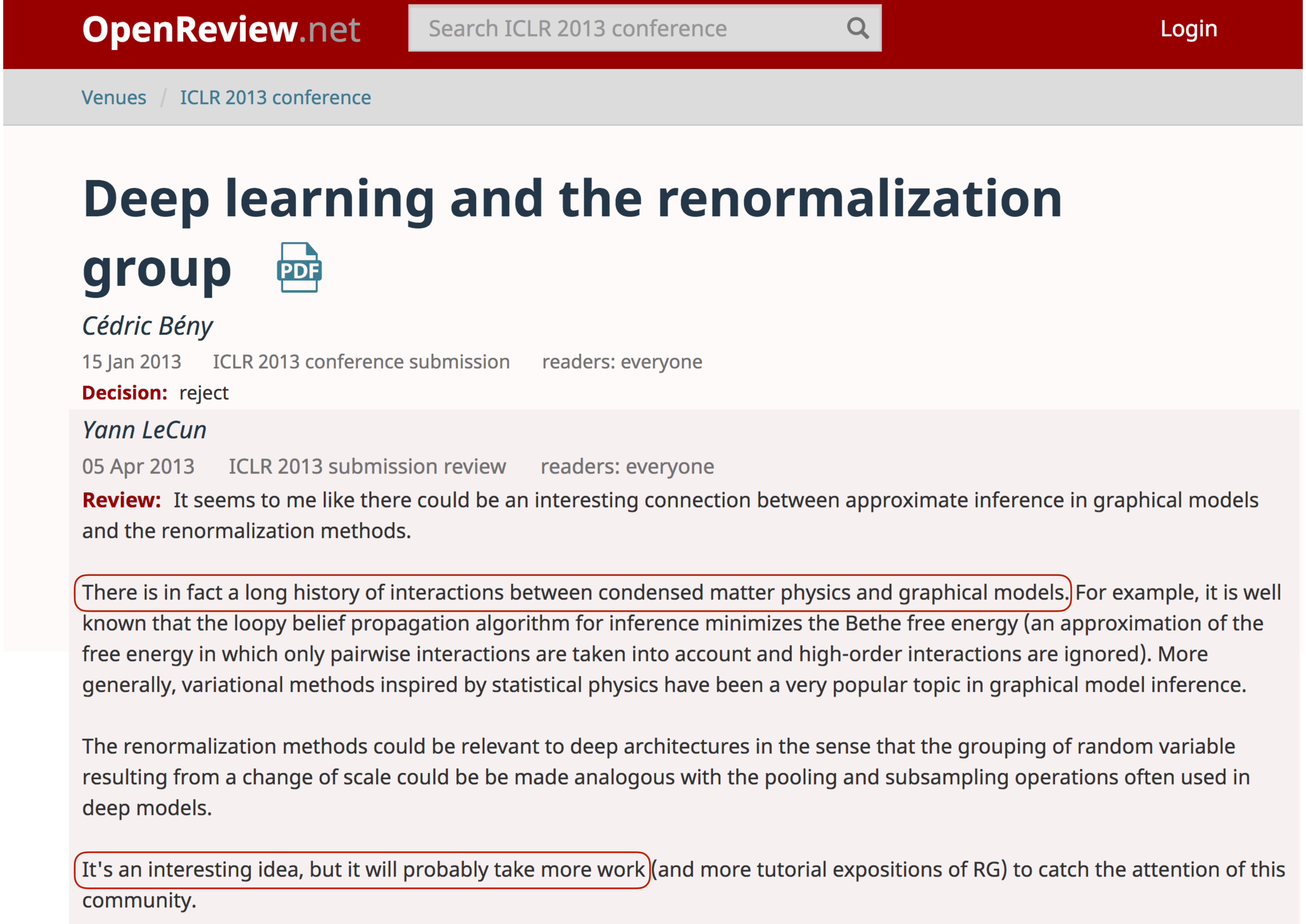
15 Jan 2013 ICLR 2013 conference submission readers: everyone

Decision: reject

Abstract: Renormalization group methods, which analyze the way in which the effective behavior of a system depends on the scale at which it is observed, are key to modern condensed-matter theory and particle physics. The aim of this paper is to compare and contrast the ideas behind the renormalization group (RG) on the one hand and deep machine learning on the other, where depth and scale play a similar role. In order to illustrate this connection, we review a recent numerical method based on the RG---the multiscale entanglement renormalization ansatz (MERA)---and show how it can be converted into a learning algorithm based on a generative hierarchical Bayesian network model. Under the assumption---common in physics---that the distribution to be learned is fully characterized by local correlations, this algorithm involves only explicit evaluation of probabilities, hence doing away with sampling.



arxiv:1301.3124



Deep learning and the renormalization group

Cédric Bény

15 Jan 2013 ICLR 2013 conference submission readers: everyone

Decision: reject

Yann LeCun

05 Apr 2013 ICLR 2013 submission review readers: everyone

Review: It seems to me like there could be an interesting connection between approximate inference in graphical models and the renormalization methods.

There is in fact a long history of interactions between condensed matter physics and graphical models. For example, it is well known that the loopy belief propagation algorithm for inference minimizes the Bethe free energy (an approximation of the free energy in which only pairwise interactions are taken into account and high-order interactions are ignored). More generally, variational methods inspired by statistical physics have been a very popular topic in graphical model inference.

The renormalization methods could be relevant to deep architectures in the sense that the grouping of random variable resulting from a change of scale could be be made analogous with the pooling and subsampling operations often used in deep models.

It's an interesting idea, but it will probably take more work (and more tutorial expositions of RG) to catch the attention of this community.

QUANTA MAGAZINE illuminating science

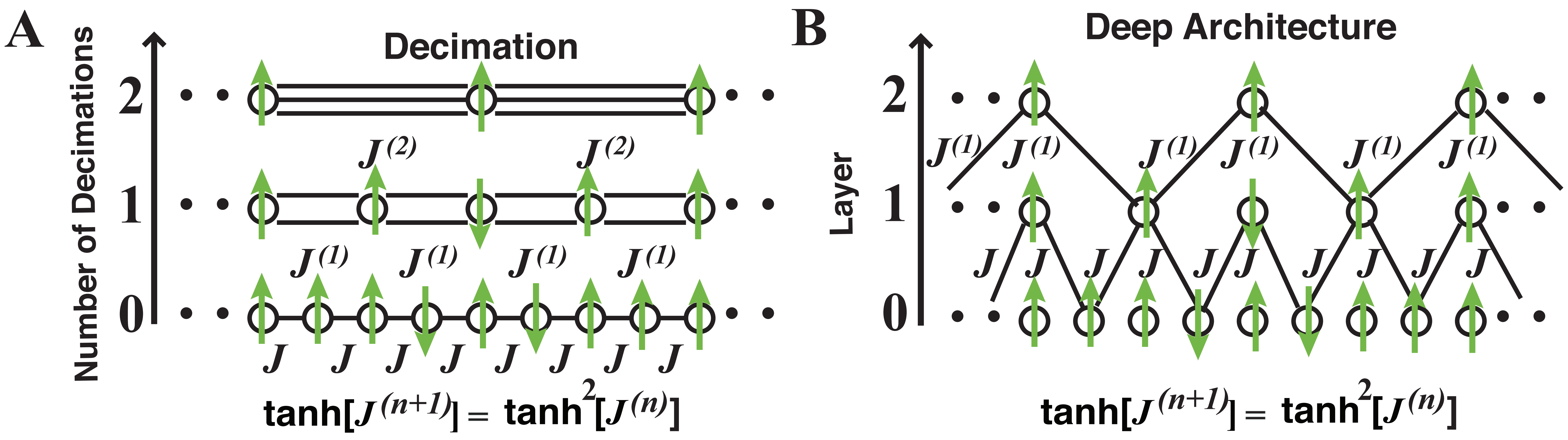
A Common Logic to Seeing Cats and Cosmos

Olena Shmahalo / Quanta Magazine

There may be a universal logic to how physicists, computers and brains tease out important features from among other irrelevant bits of data.

“An exact mapping between the Variational Renormalization Group and Deep Learning”, Mehta and Schwab, 1410.3831

Exact Mapping



$$e^{-E(\boldsymbol{h})} = \sum_{\boldsymbol{x}} e^{T(\boldsymbol{x},\boldsymbol{h})-E(\boldsymbol{x})}$$

$$e^{-E(\boldsymbol{h})} = \sum_{\boldsymbol{x}} e^{-E(\boldsymbol{x},\boldsymbol{h})}$$

RG Transformation

Boltzmann Machine

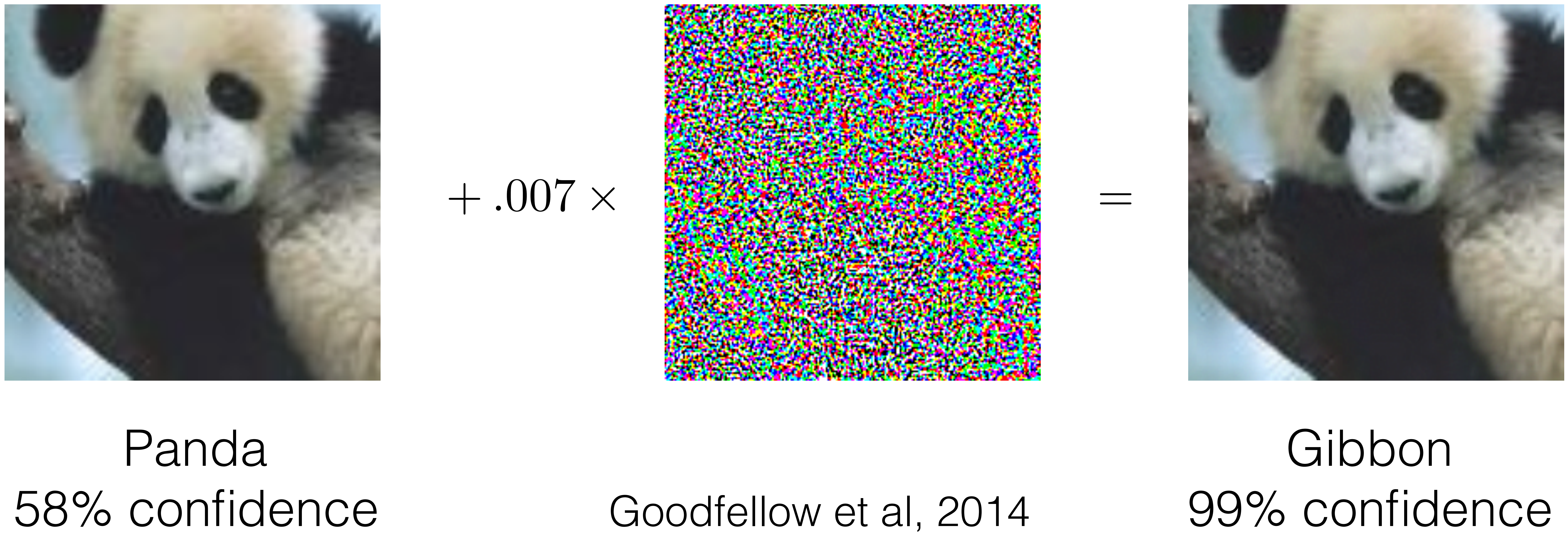
More on DL and RG

- “Why does deep and cheap learning work so well ”, Lin, Tegmark, Rolnick, 1608.08225
- Comment on the paper above, Schwab and Mehta,1609.03541
- PCA meets RG, Bradde and Bialek, 1610.09733
- Mutual information RG, Koch-Janusz and Ringel, 1704.06279

next talk
by Maciej

- Machine Learning Holography, You, Yang, Qi, 1709.01223
- Vulnerability of deep learning, Kenway, 1803.06111 & 1803.10995

More on DL and RG



Goodfellow et al, 2014

- Vulnerability of deep learning, Kenway, 1803.06111 & 1803.10995

Why bother ?

RG offers a theoretical understanding of DL

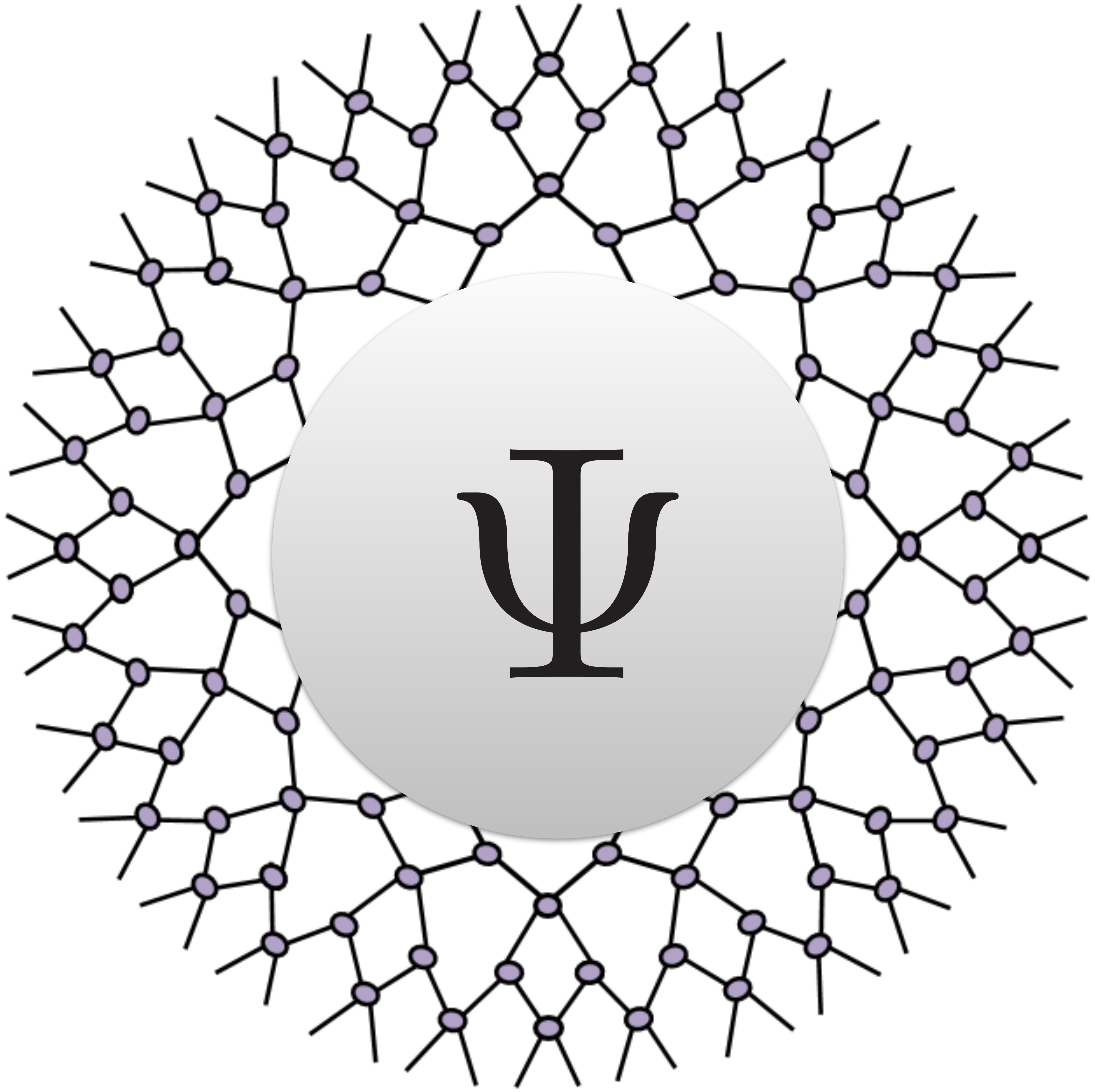
In return, DL helps to solve physics problems

Shuo-Hui Li (李烁辉)

arXiv:1802.02840

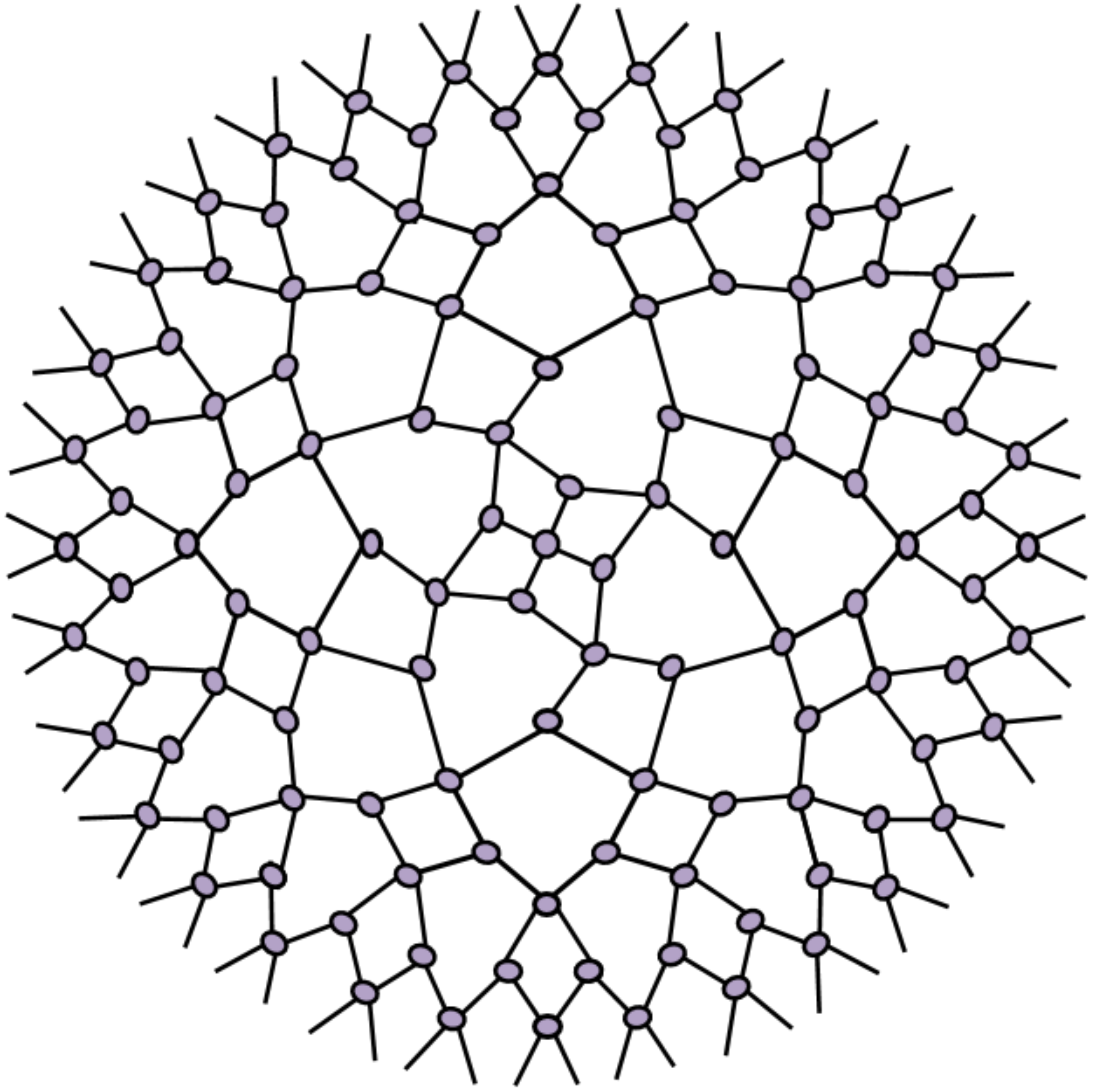
https://github.com/li012589/NeuralRG

Multi-Scale Entanglement Renormalization Ansatz



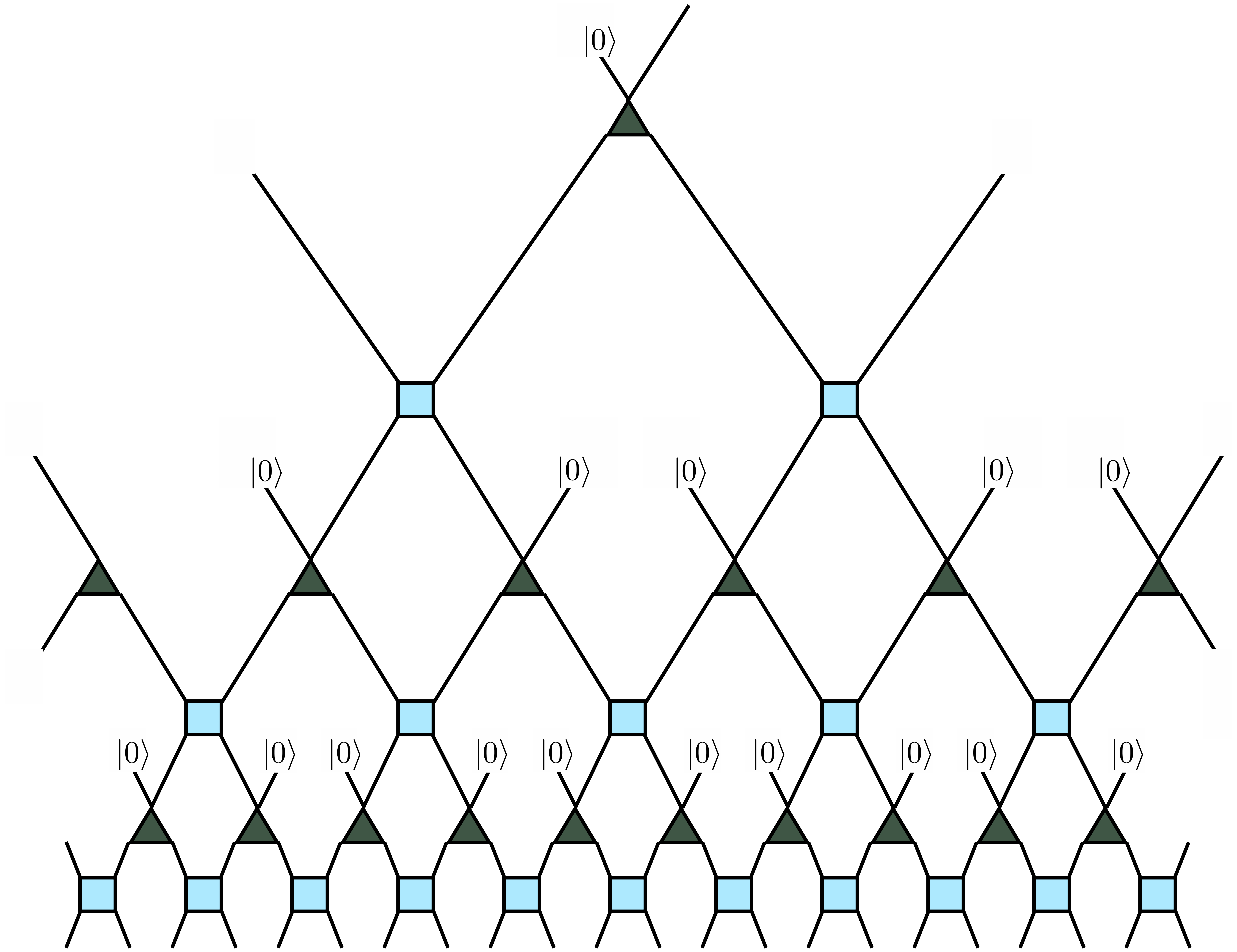
Vidal 2006

Multi-Scale Entanglement Renormalization Ansatz



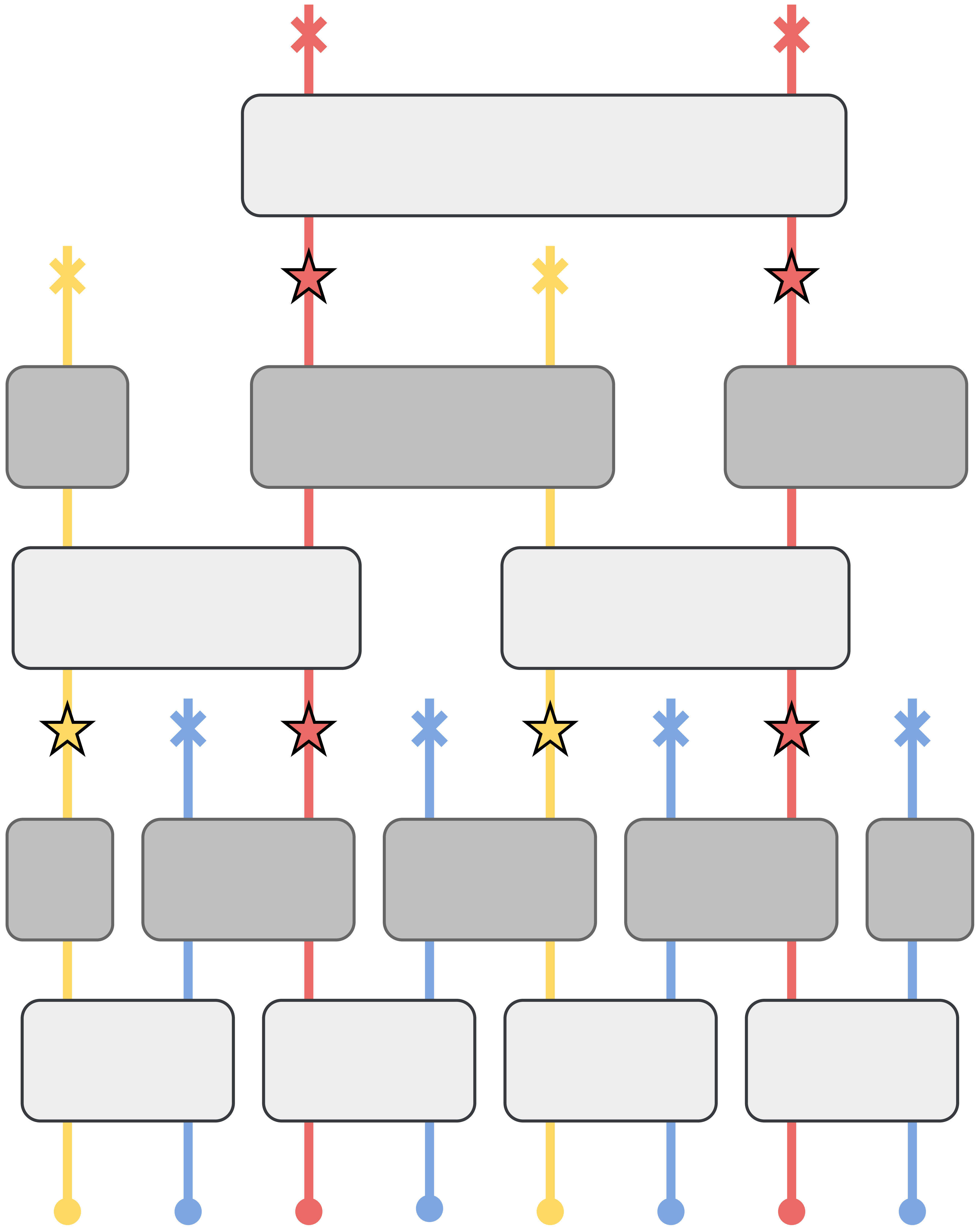
Vidal 2006

MERA as a quantum circuit



Entangled qubits

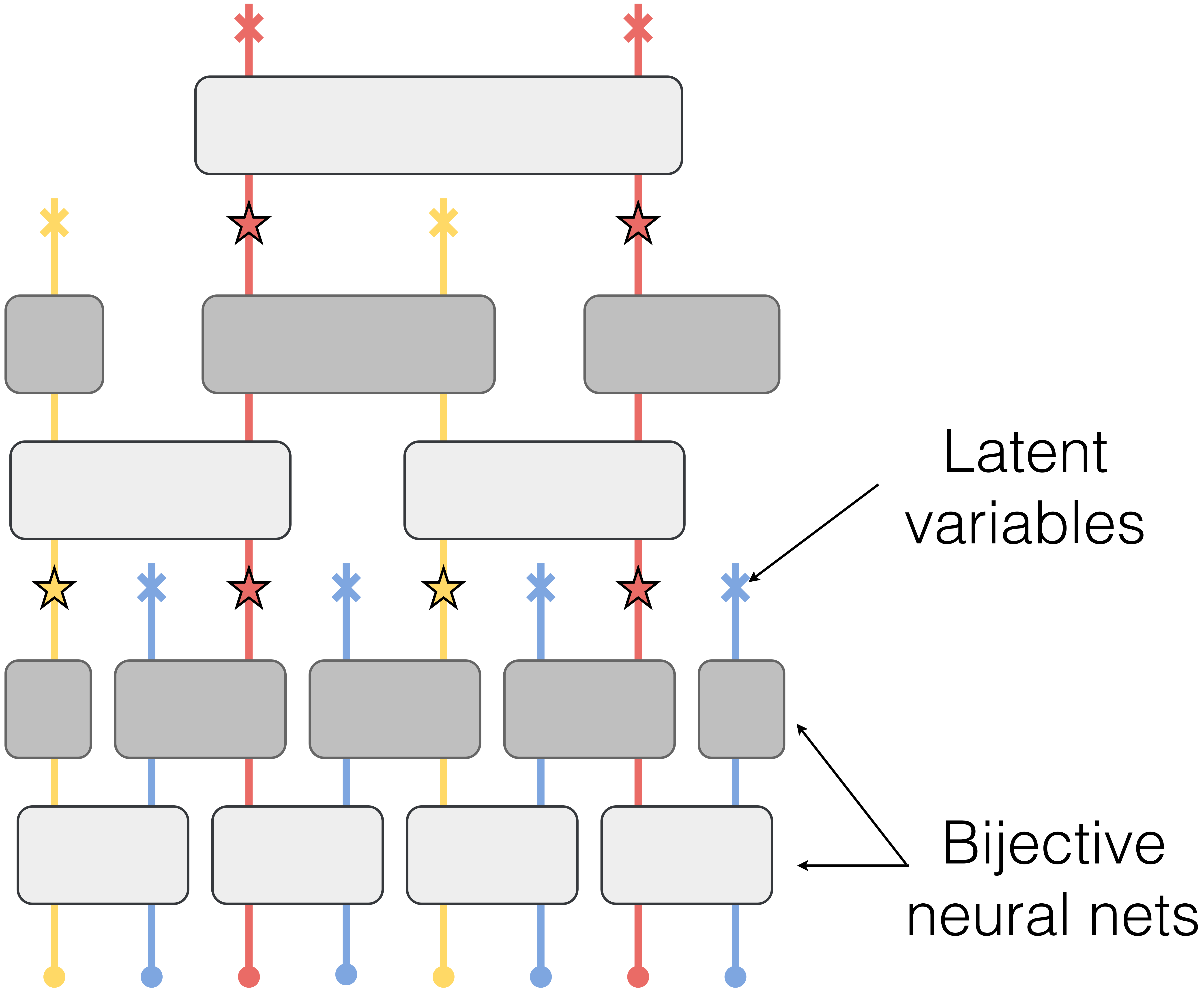
Neural Network Renormalization Group



Bijective neural nets

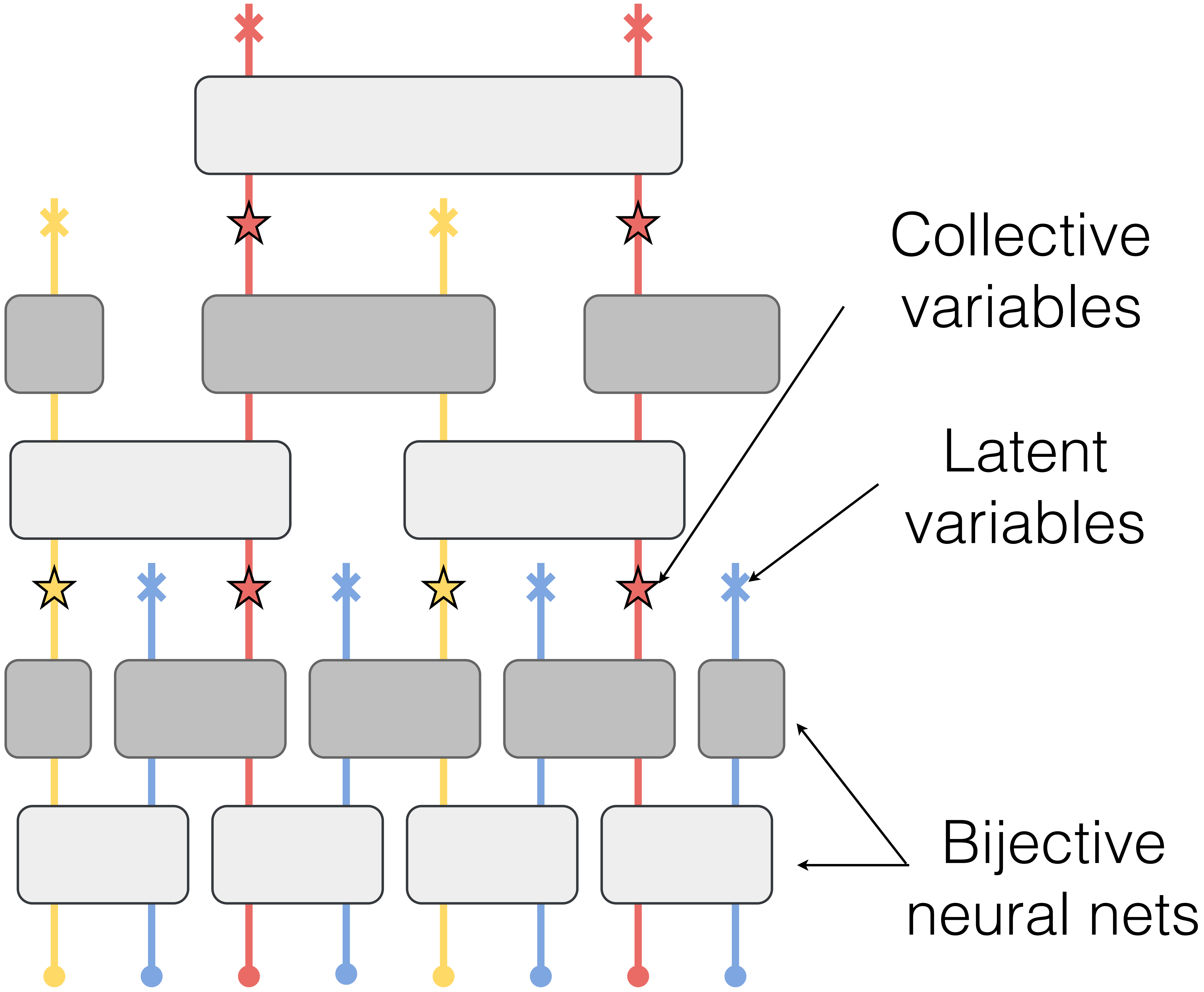
Correlated classical variables

Neural Network Renormalization Group



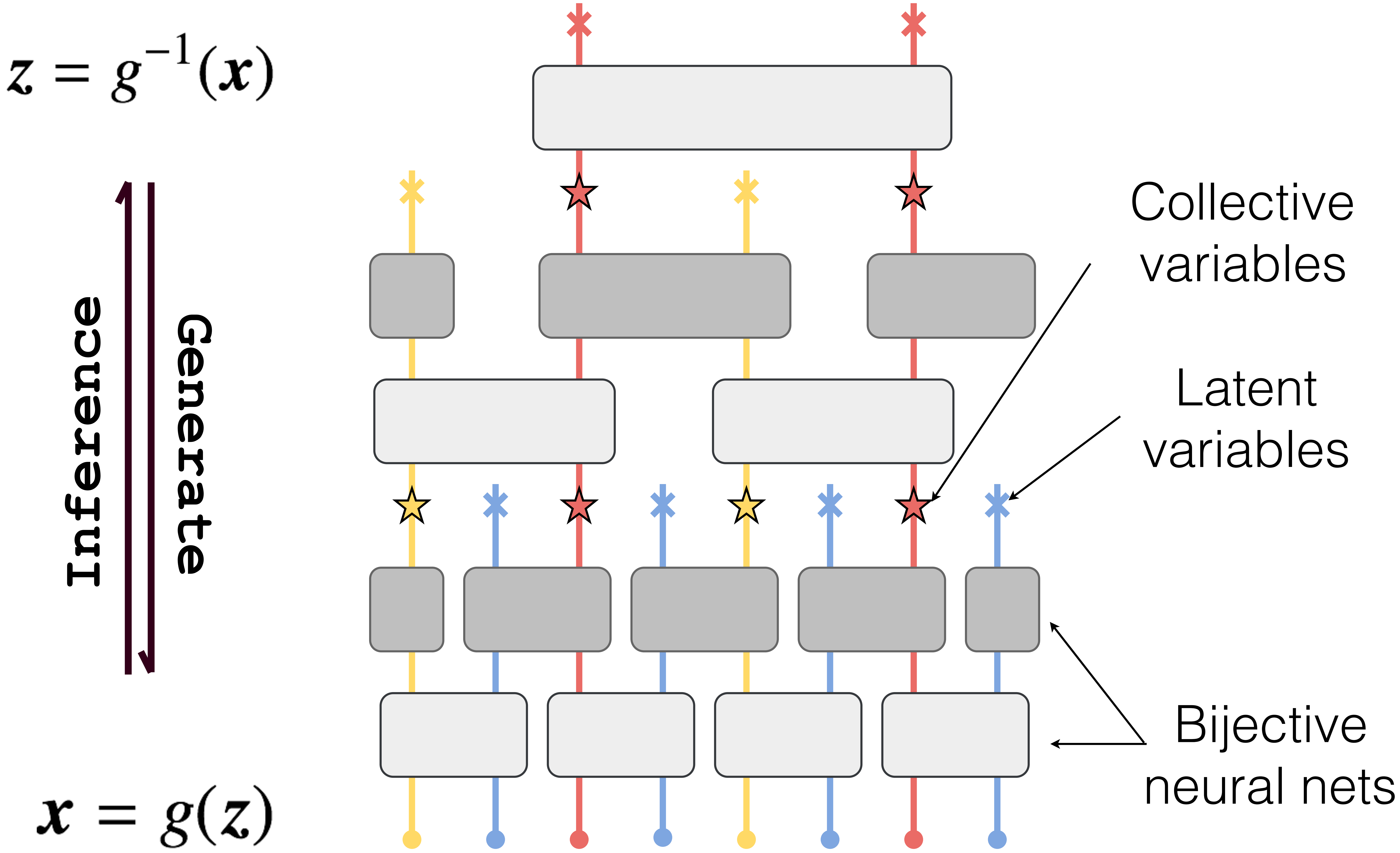
Correlated classical variables

Neural Network Renormalization Group



Correlated classical variables

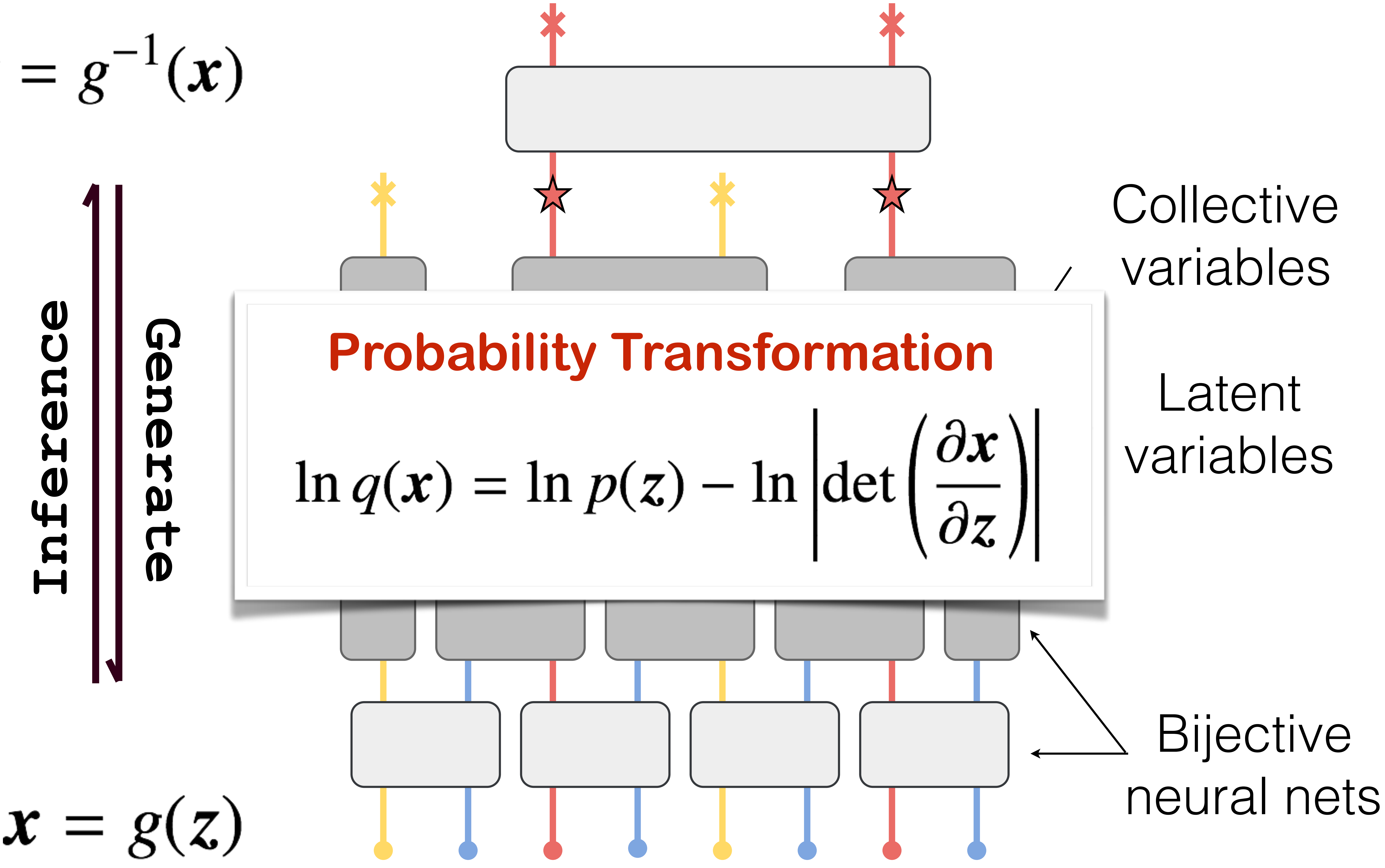
Neural Network Renormalization Group



Correlated classical variables

Neural Network Renormalization Group

$$z = g^{-1}(x)$$



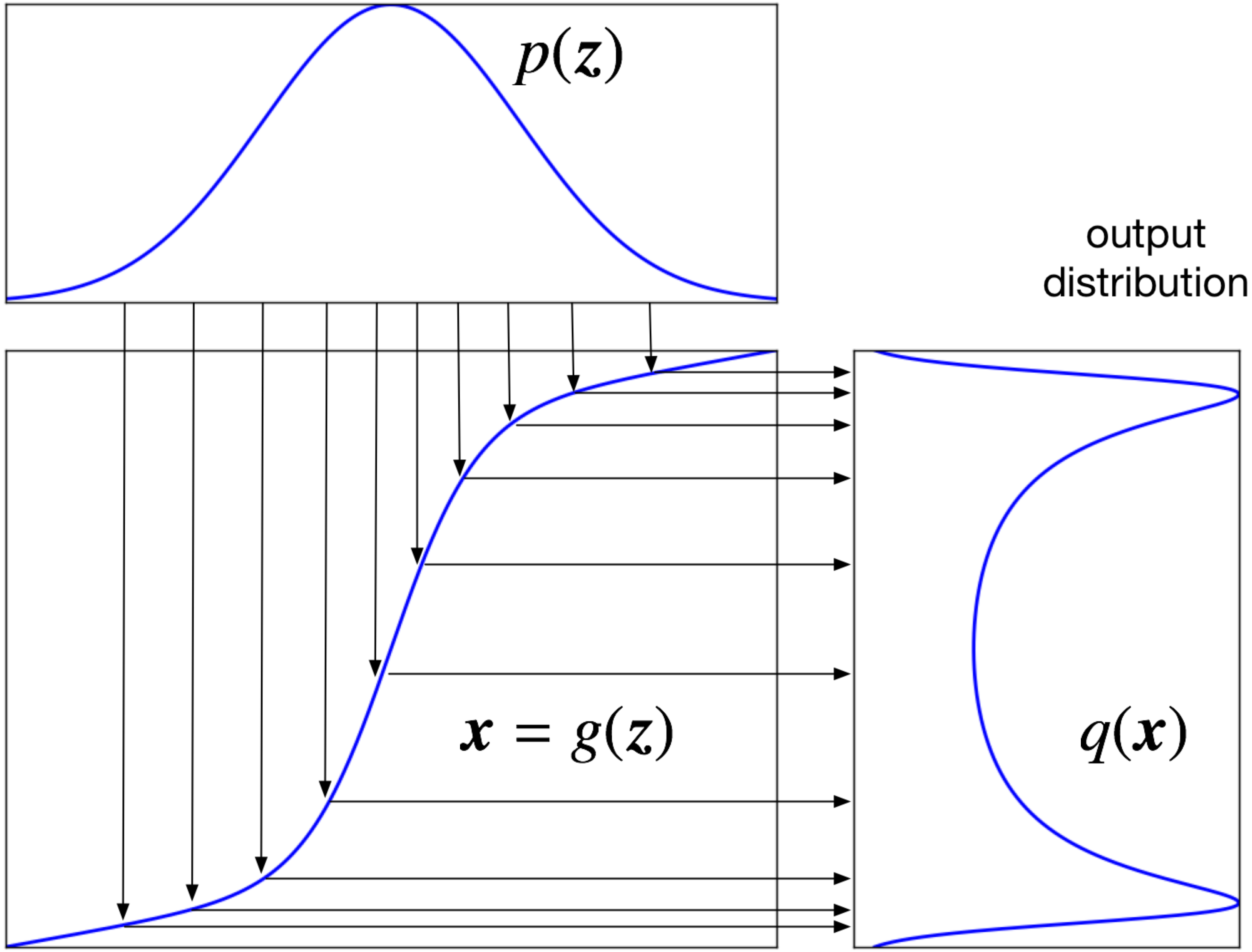
Probability Transformation

$$\ln q(x) = \ln p(z) - \ln \left|\det\left(\frac{\partial x}{\partial z}\right)\right|$$

$$x = g(z)$$

Correlated classical variables

Probability transformation in picture

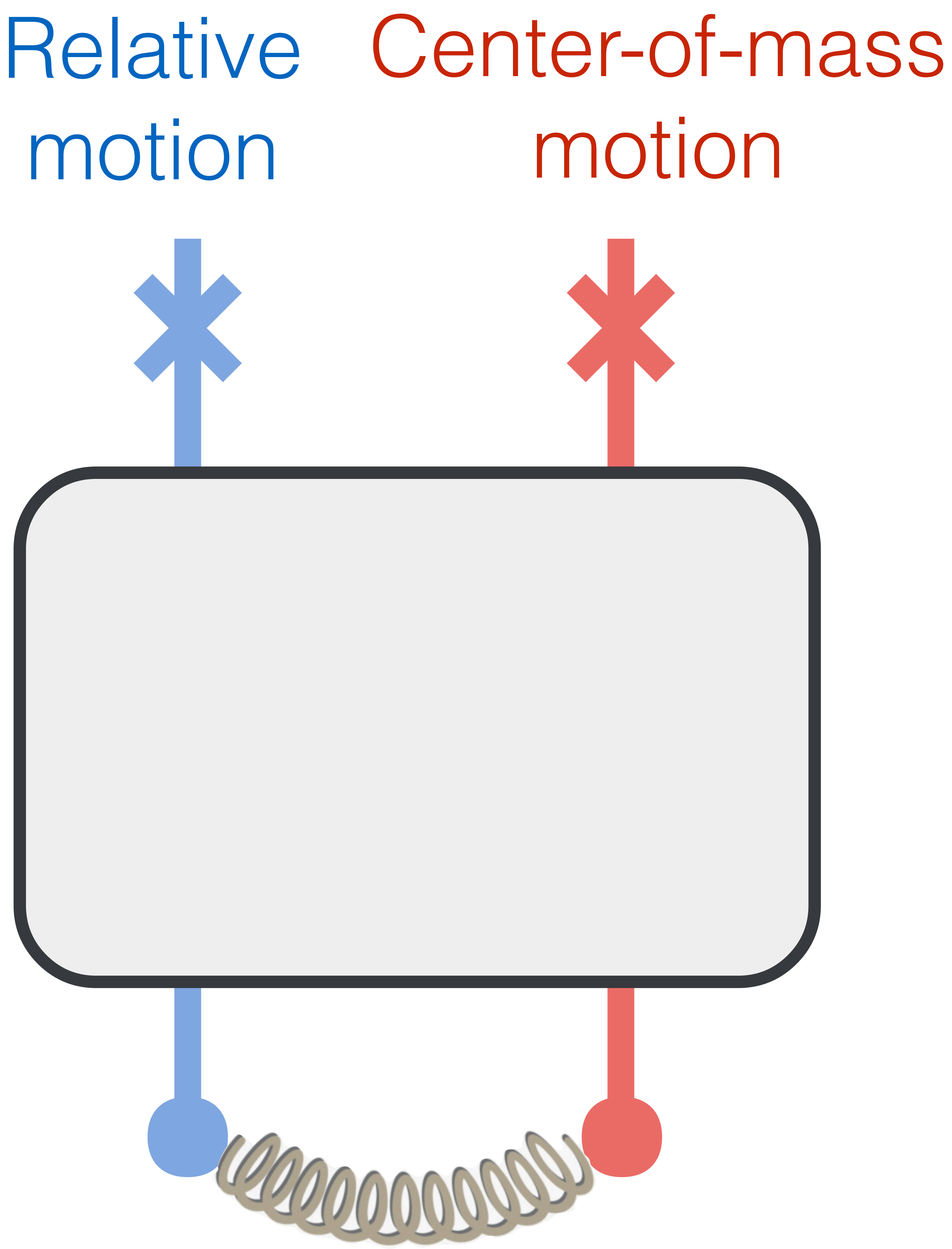


input distribution

function computed by the network

output distribution

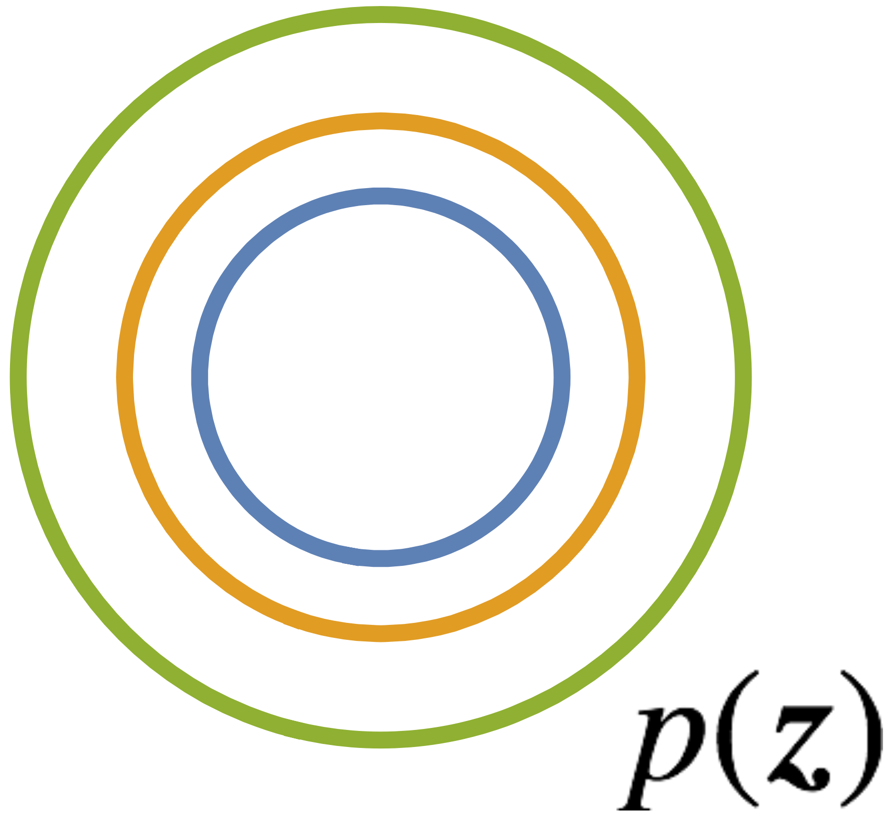
Toy problem: Harmonic oscillator

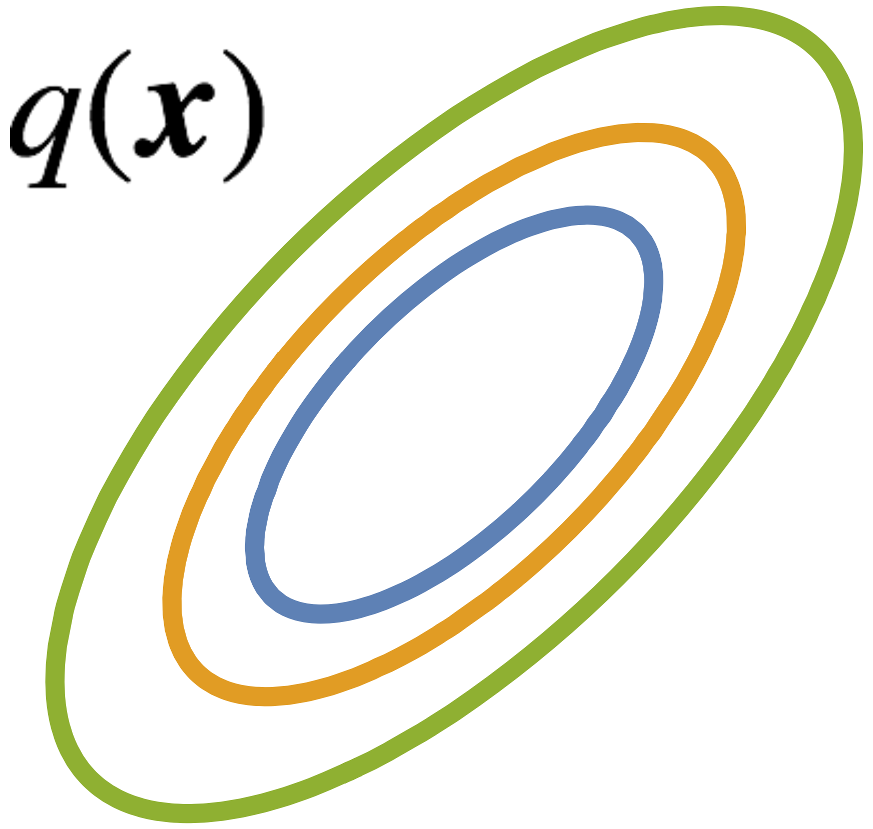


Relative motion

Center-of-mass motion

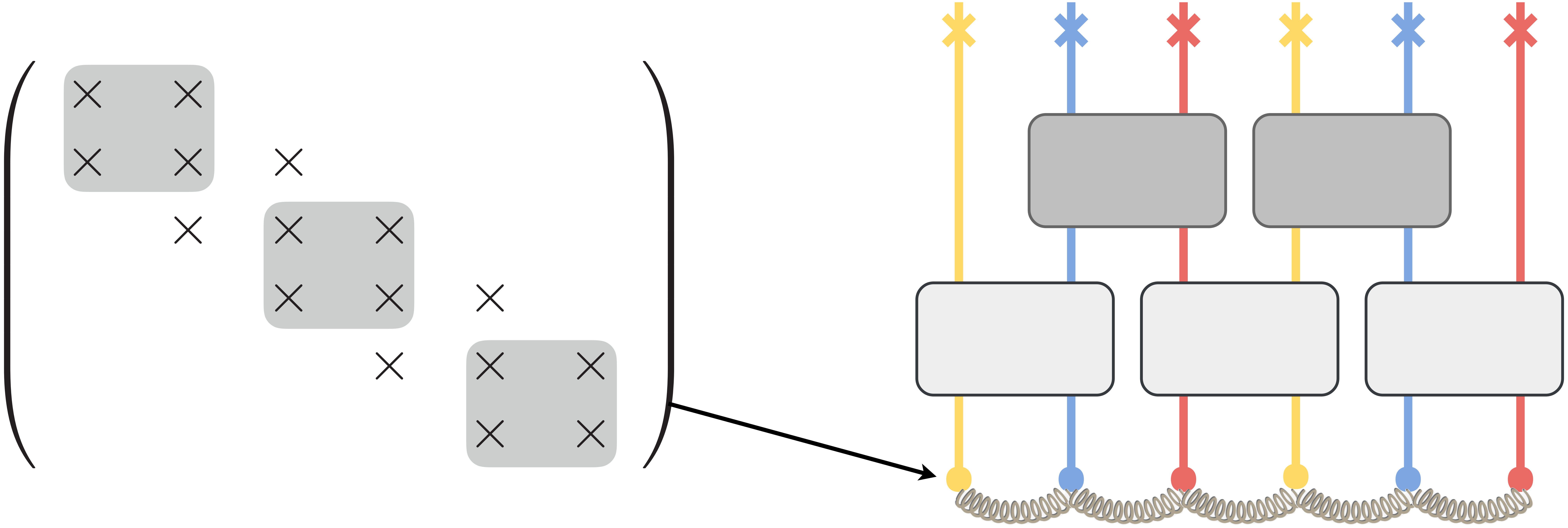
Coupled harmonic oscillator





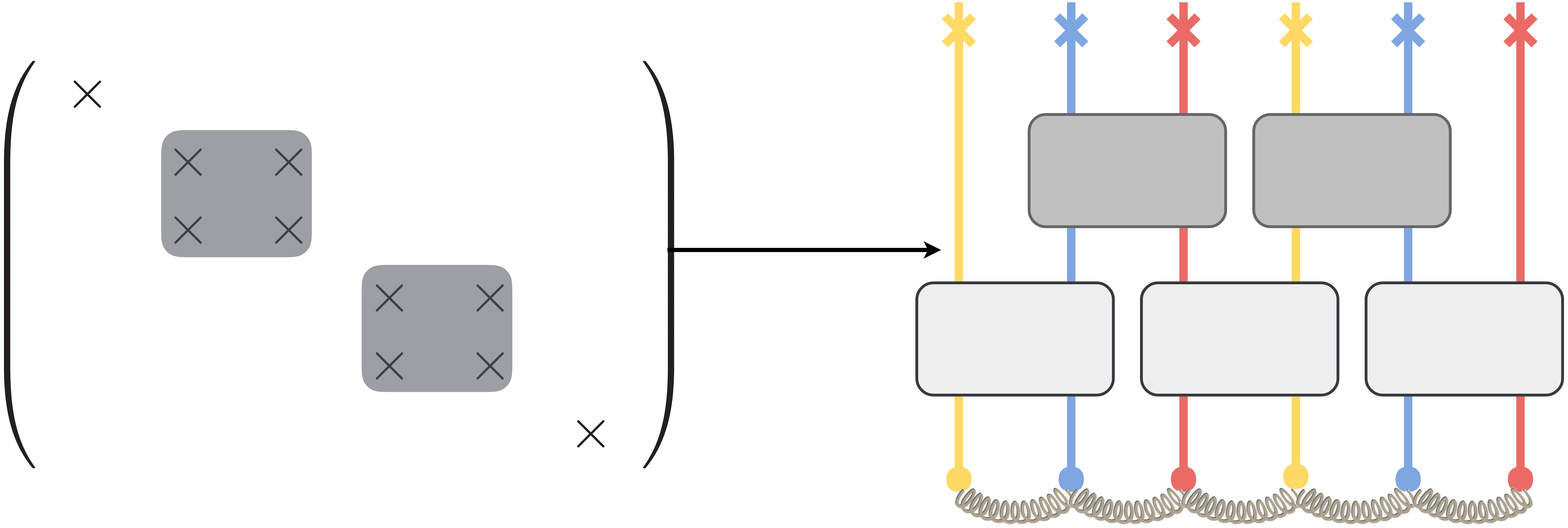
Toy problem: Harmonic oscillator chain

Linear layers are sufficient to decouple a free theory via iterative diagonalization



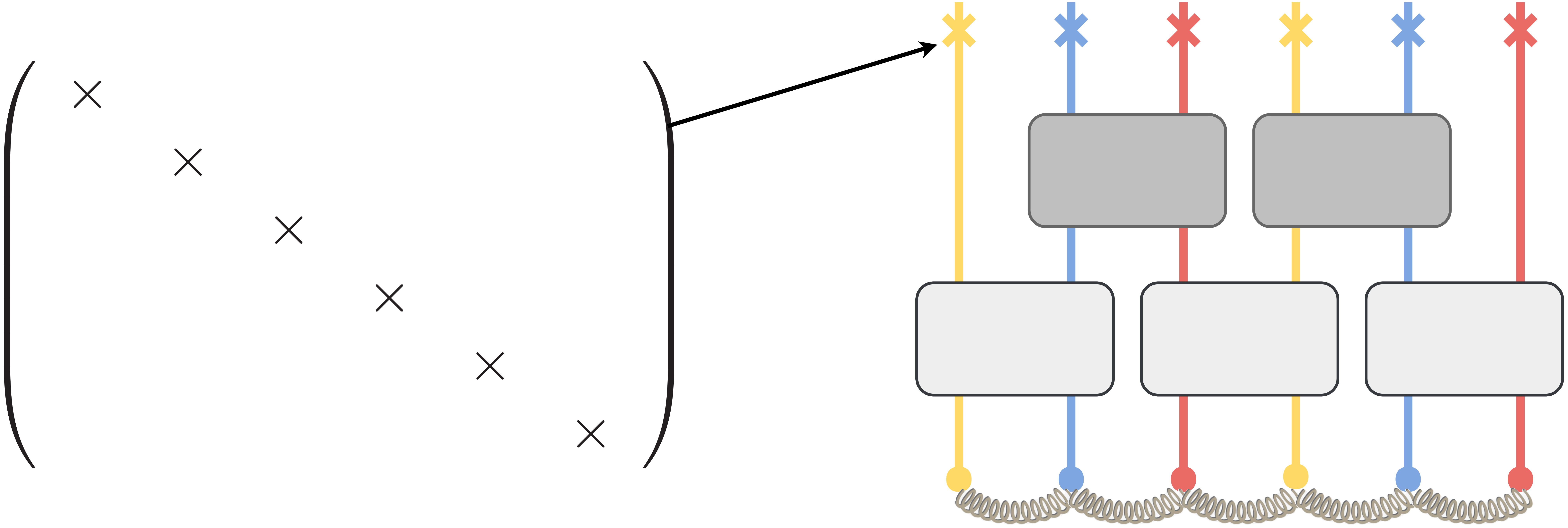
Toy problem: Harmonic oscillator chain

Linear layers are sufficient to decouple a free theory via iterative diagonalization



Toy problem: Harmonic oscillator chain

Linear layers are sufficient to decouple a free theory via iterative diagonalization



Nonlinear Bijectors

Bijective & Differentiable map, i.e., Diffeomorphism

Forward

Arbitrary neural nets

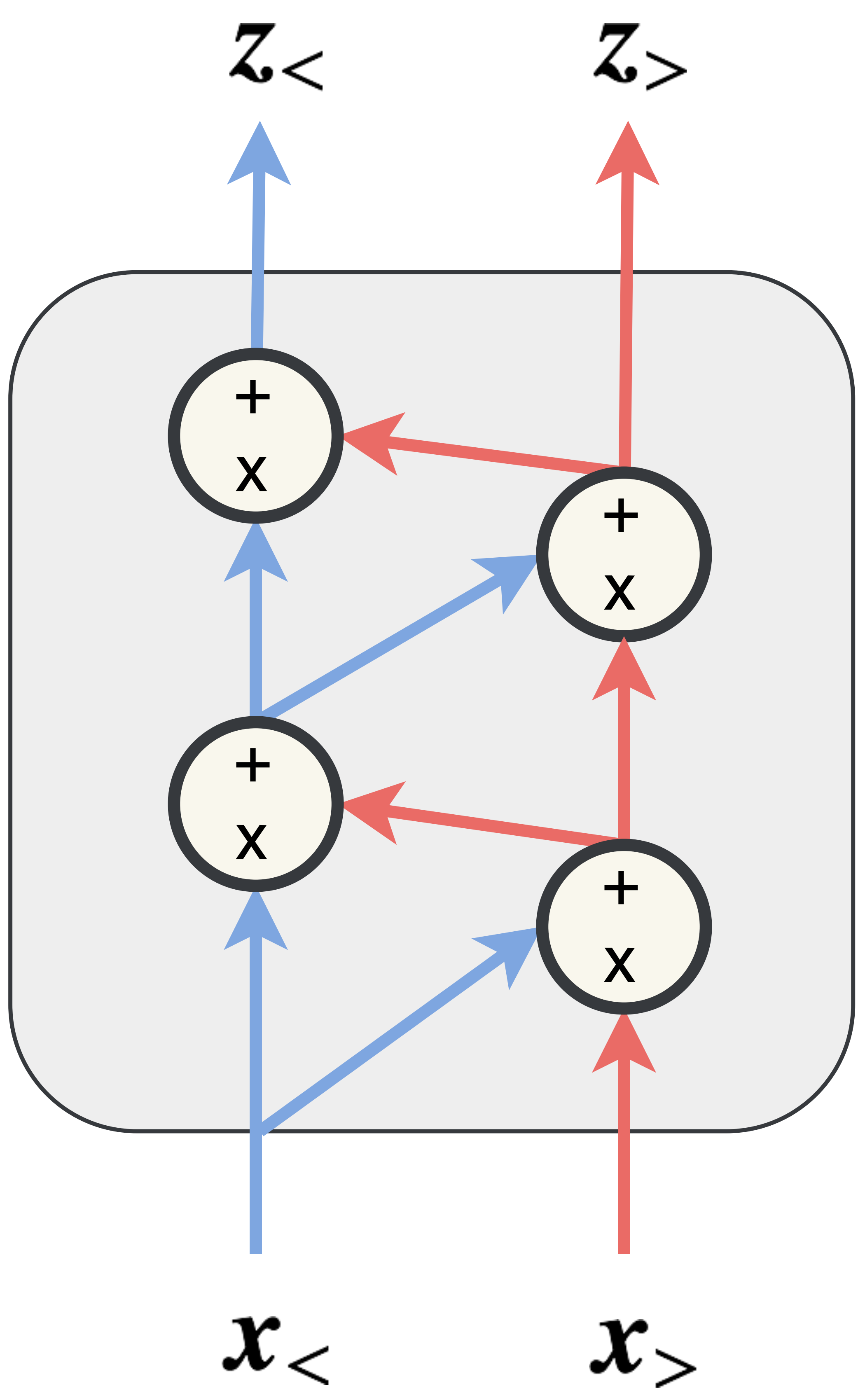
$$\begin{cases} \boldsymbol{x}_< = \boldsymbol{z}_< \\ \boldsymbol{x}_> = \boldsymbol{z}_> \odot e^{s(\boldsymbol{z}_<)} + t(\boldsymbol{z}_<) \end{cases}$$

Inverse

$$\begin{cases} \boldsymbol{z}_< = \boldsymbol{x}_< \\ \boldsymbol{z}_> = (\boldsymbol{x}_> - t(\boldsymbol{x}_<)) \odot e^{-s(\boldsymbol{x}_<)} \end{cases}$$

Log-Abs-Jacobian-Det

$$\ln \left| \det \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} \right) \right| = \sum_i [s(\boldsymbol{z}_<)]_i$$



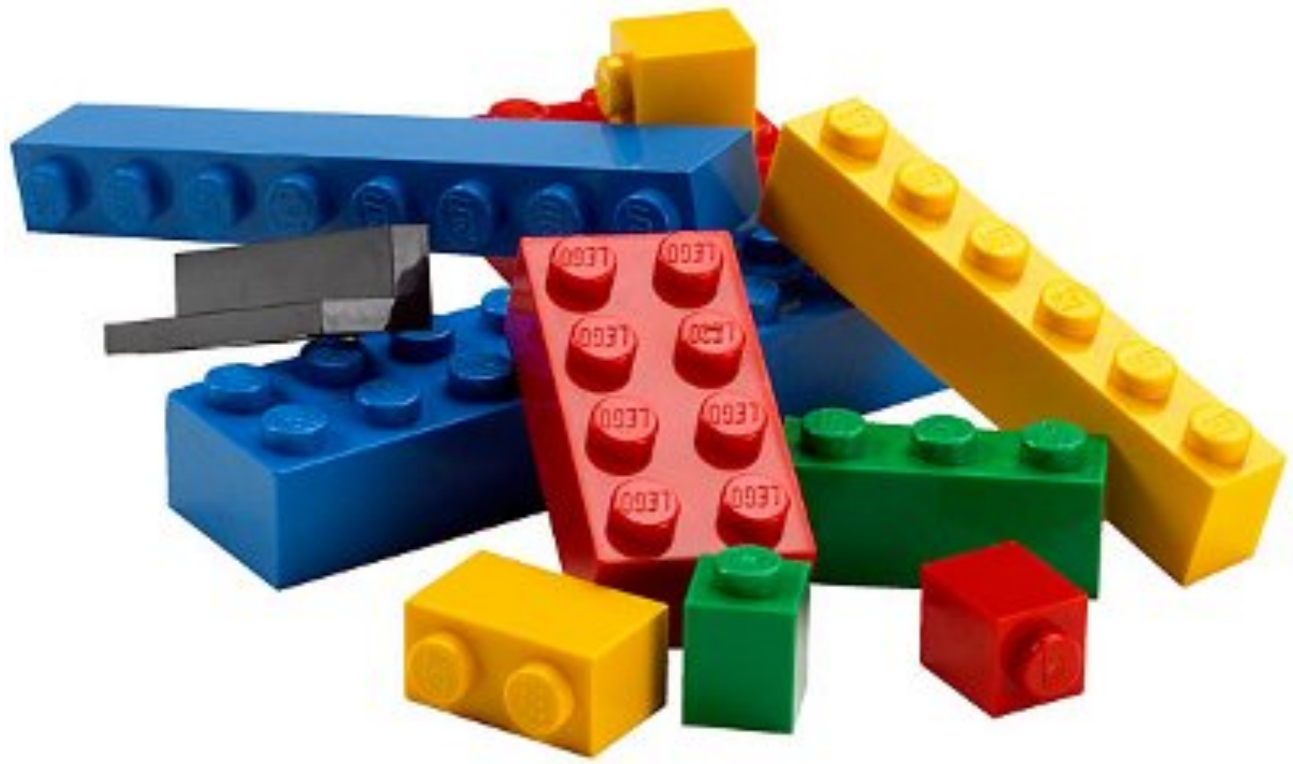
Normalizing flow, Rezende et al,1505.05770
Real NVP, Dinh et al,1605.08803

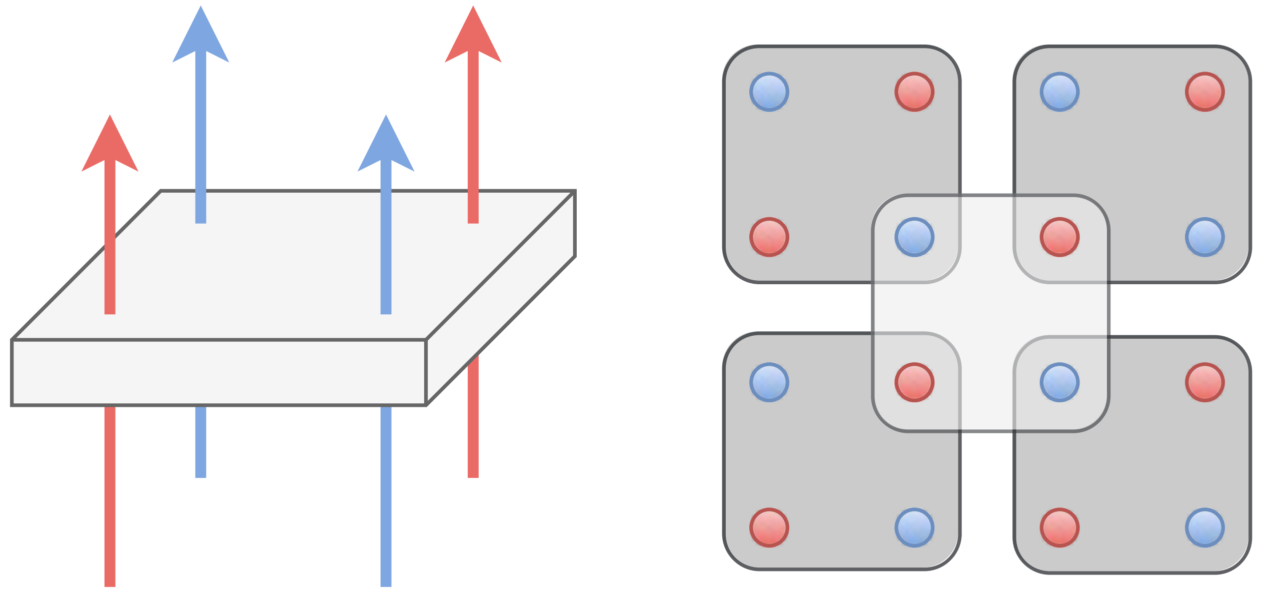
https://www.tensorflow.org/api_docs/python/tf/distributions/bijectors/Bijector
http://pytorch.org/docs/master/distributions.html#transformeddistribution

Bijectors form a group

$$\boldsymbol{x} = g(\boldsymbol{z})$$
$$g = \cdots \circ g_2 \circ g_1$$

$$\ln\left|\det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}\right)\right| = \sum_i \ln\left|\det\left(\frac{\partial g_{i+1}}{\partial g_i}\right)\right|$$

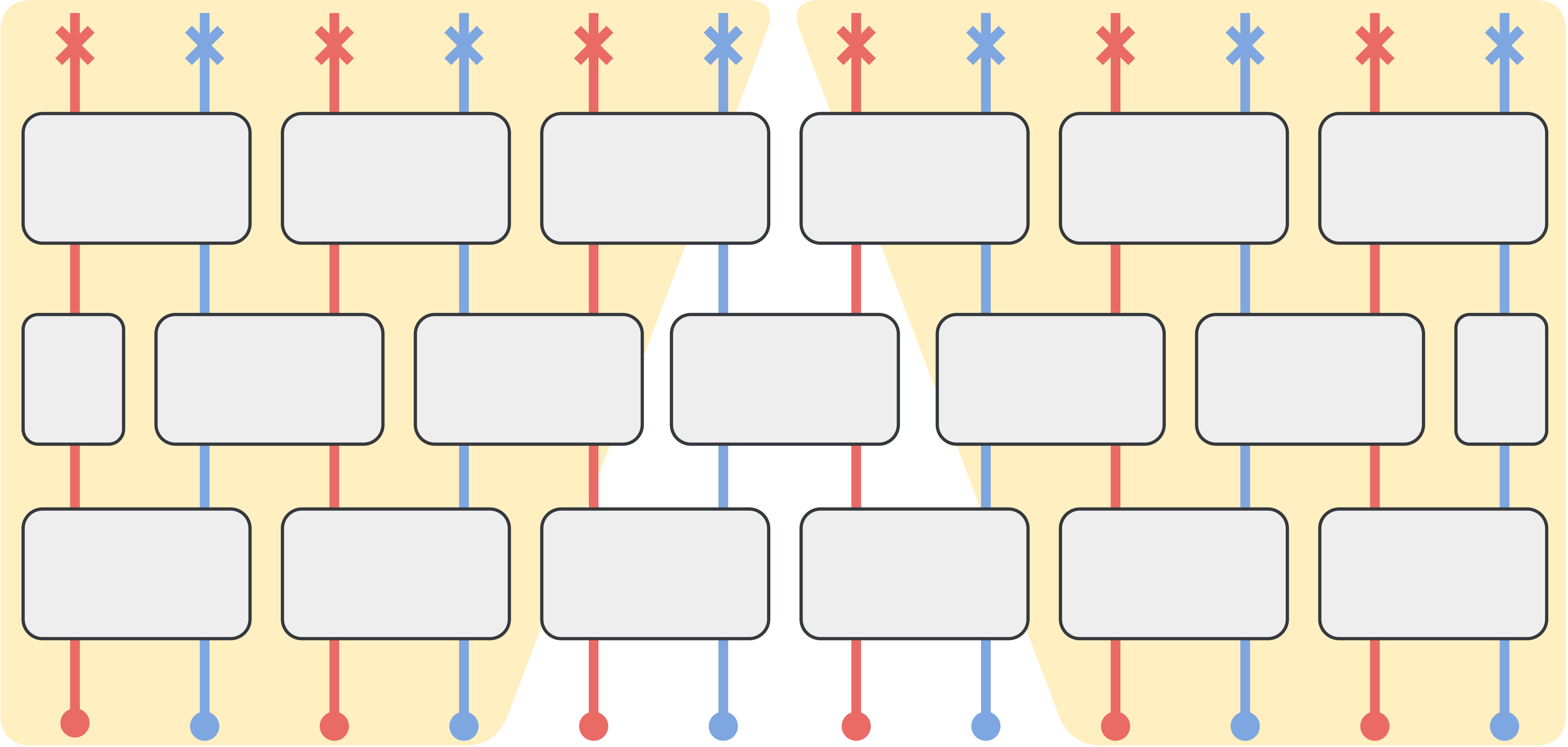




Modular design

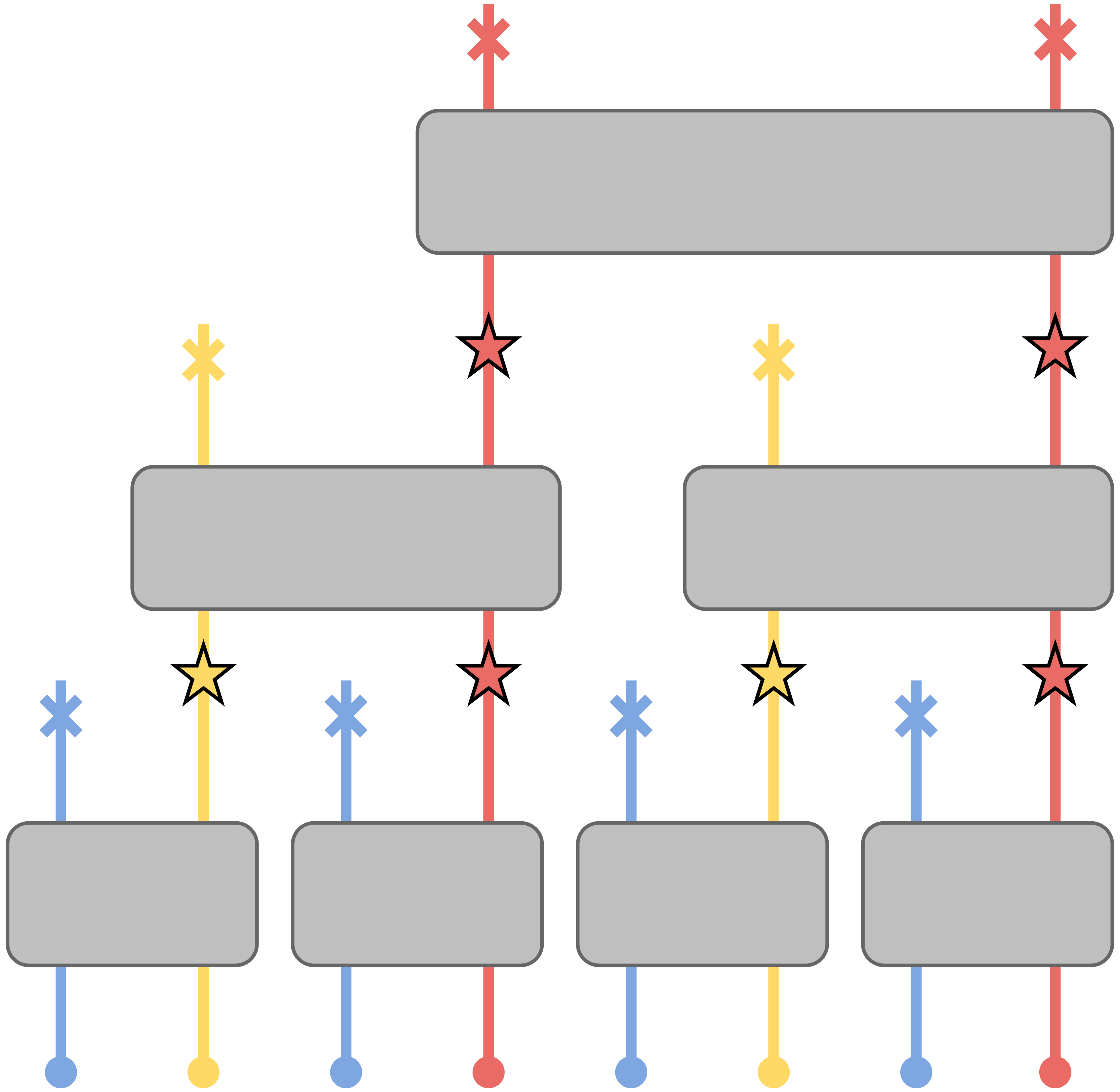
Flexible structure

“Disentangler only” architecture

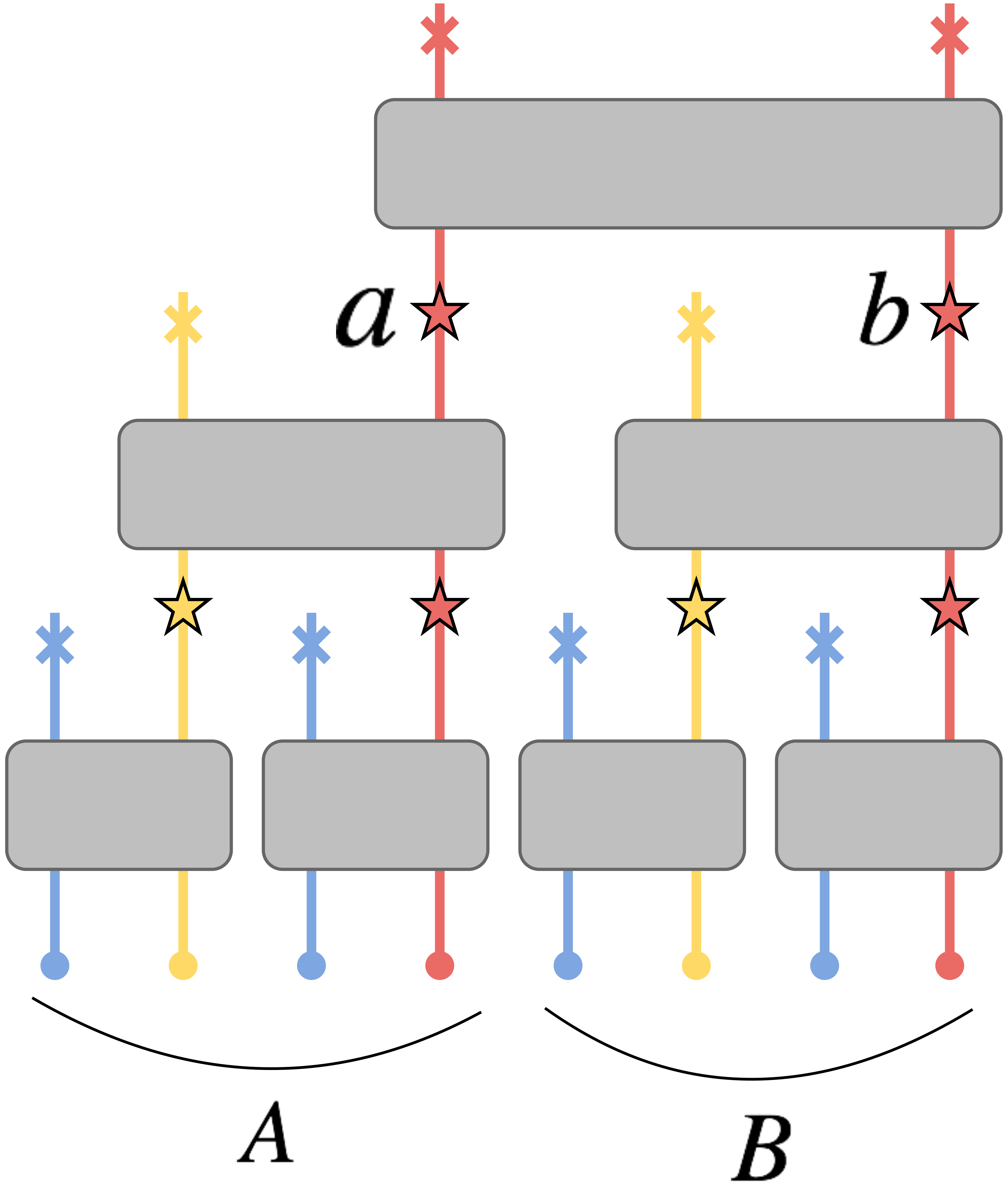


Correlation length ~ Network depth

“Decimator only” architecture



“Decimator only” architecture

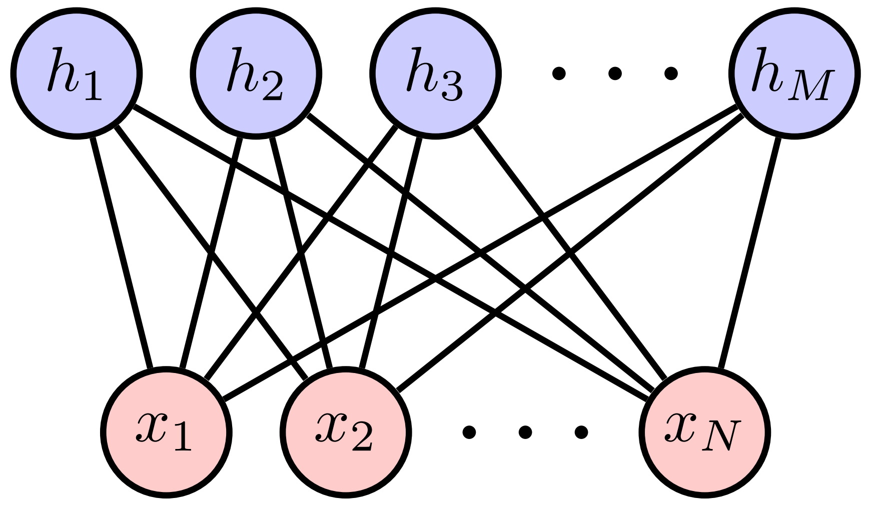


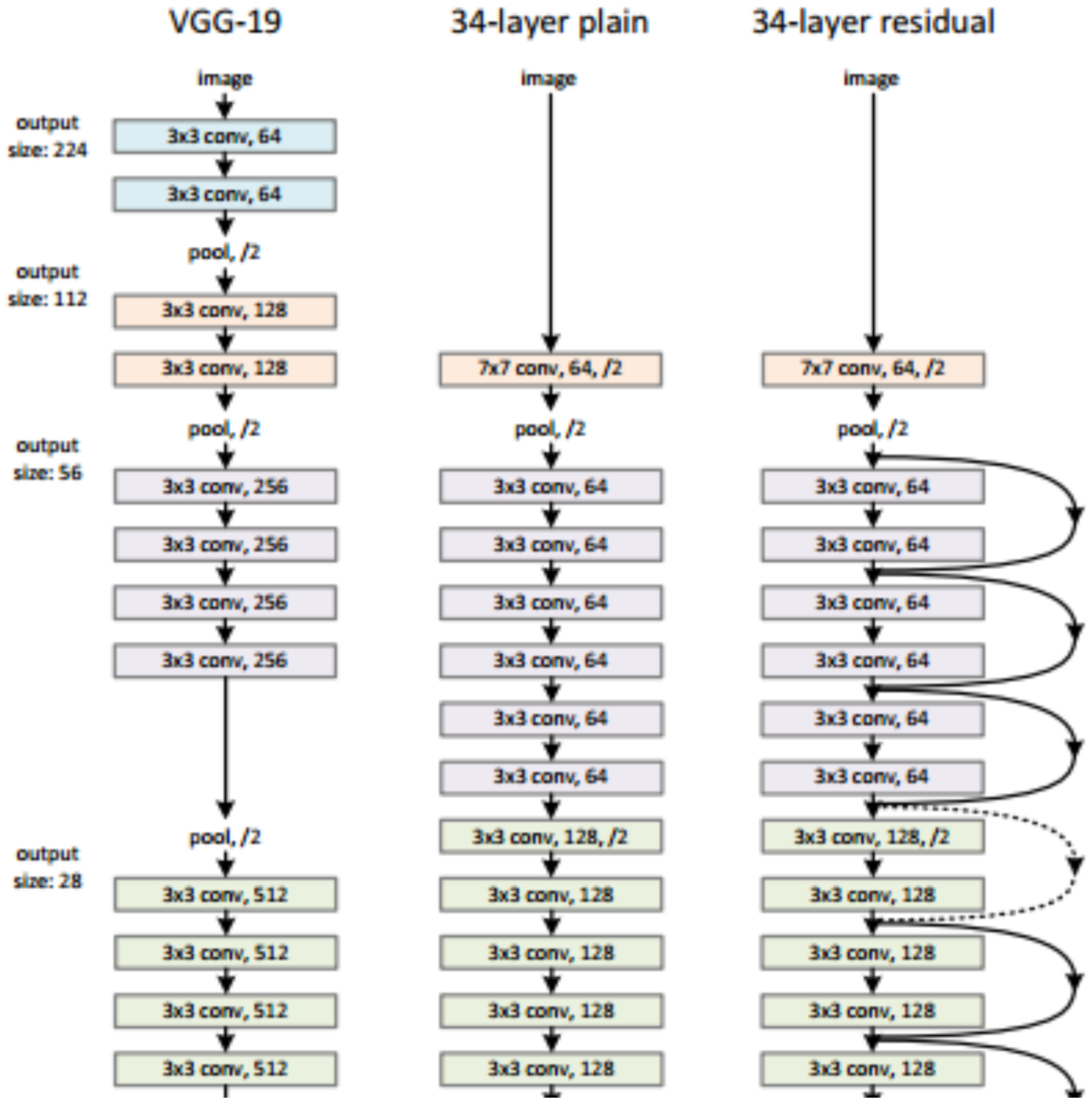
$$I(A:B) = I(a:b)$$

Mutual Information Bottleneck

Spherical chicken
in vacuum

Animals
in the wild

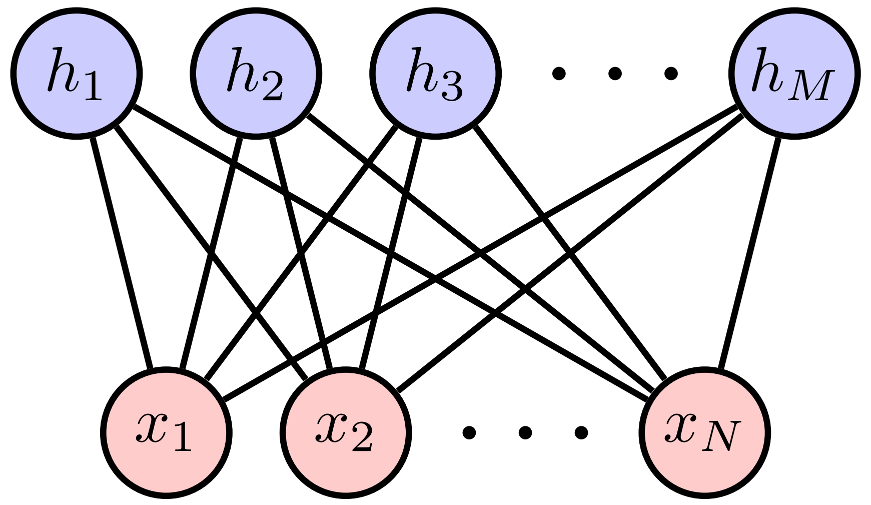


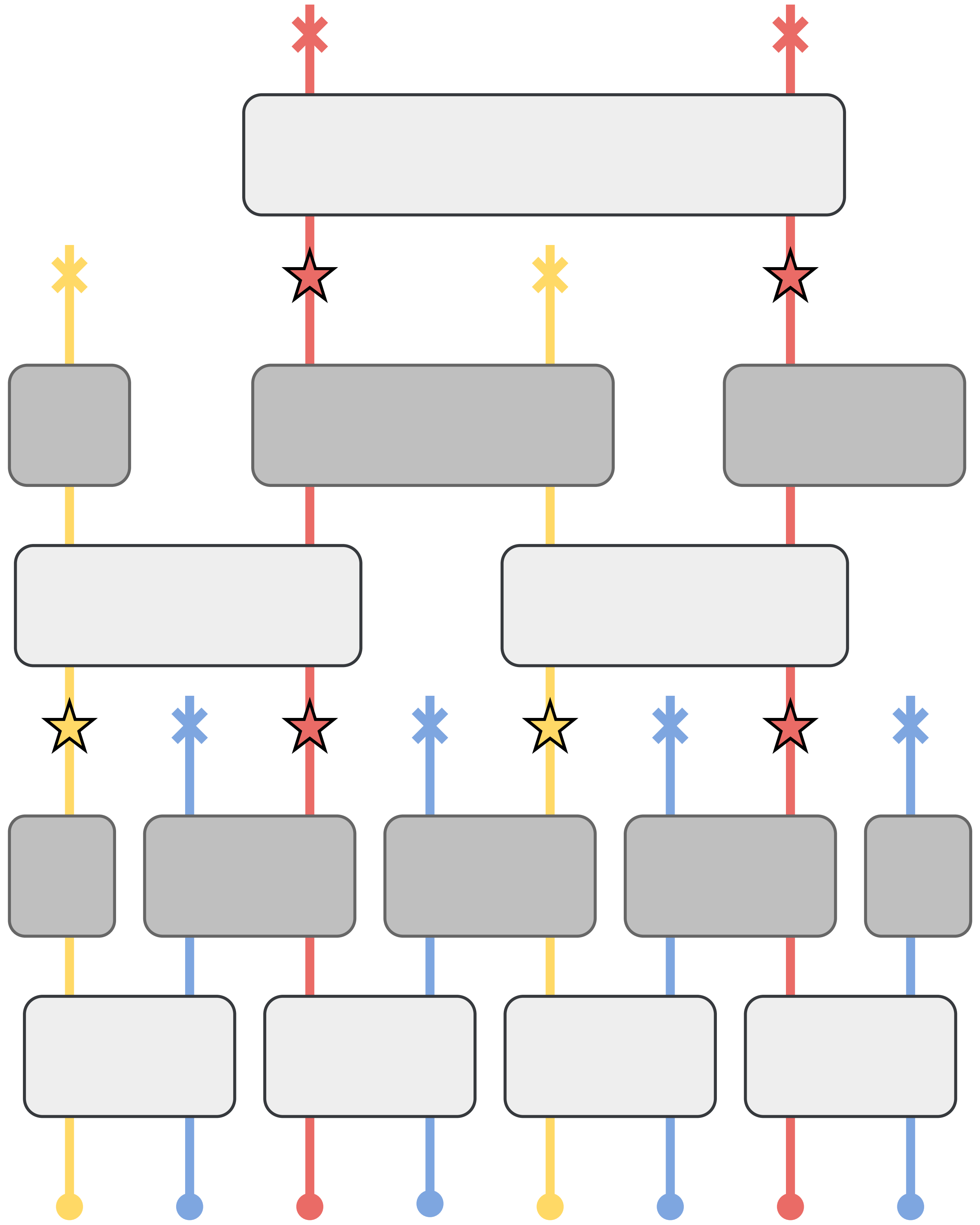


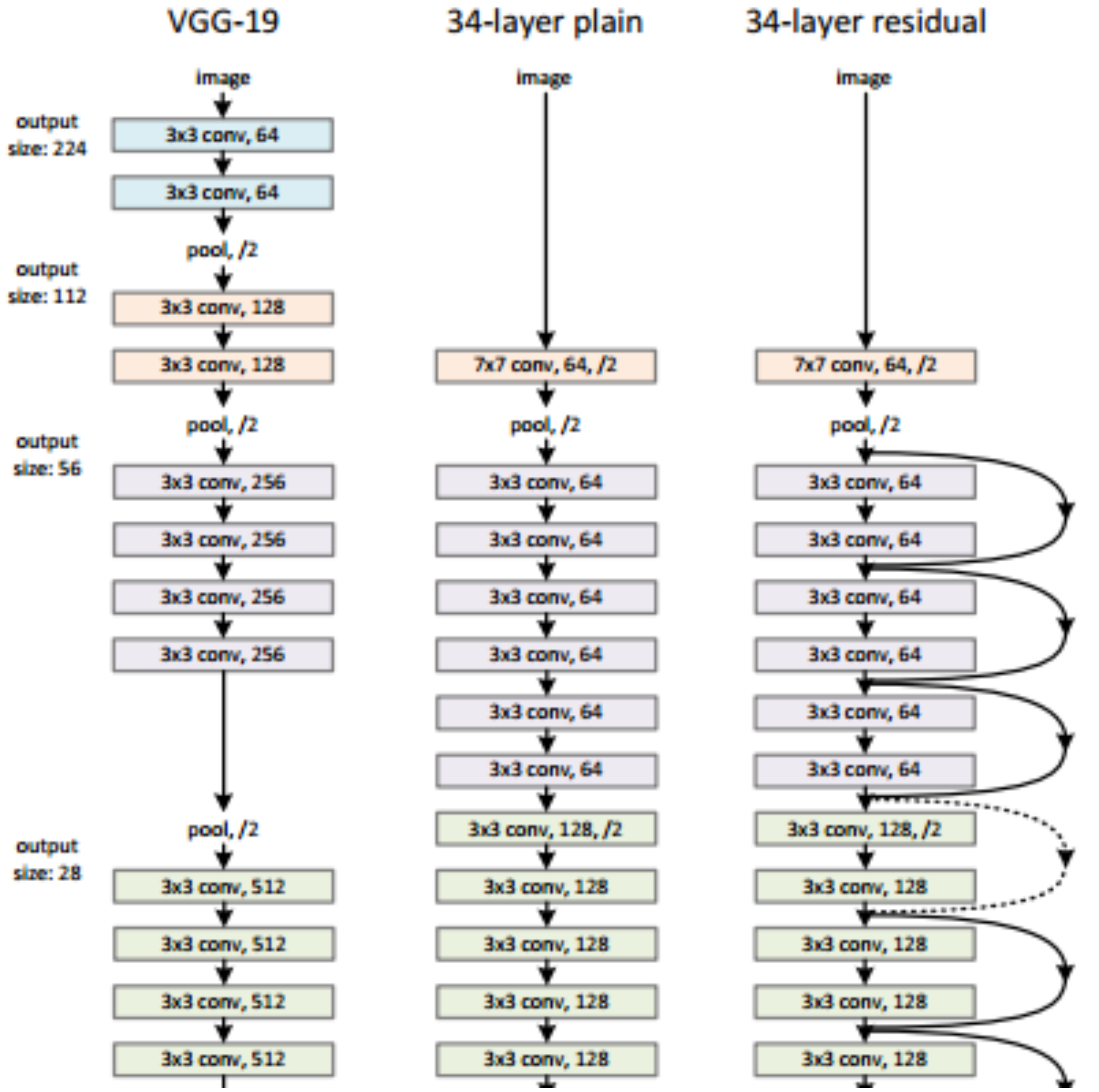
Simplified, but not oversimplified model with balanced interpretability and expressibility

Spherical chicken
in vacuum

Animals
in the wild







Training: Probability Density *Estimation ?*

Given a dataset, learn its probability density by minimizing the Negative Log-Likelihood

$$\mathrm{NLL}_{\boldsymbol{\theta}} = -\sum_{\boldsymbol{x} \in \mathrm{dataset}} \ln q_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Network parameters

Training: Probability Density *Estimation ?*

Given a dataset, learn its probability density by minimizing the Negative Log-Likelihood

$$\mathrm{NLL}_{\boldsymbol{\theta}} = -\sum_{\boldsymbol{x} \in \mathrm{dataset}} \ln q_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Network parameters

Equivalent to optimize the forward Kullback–Leibler divergence

$$\mathbb{KL}\left(\frac{e^{-E(\boldsymbol{x})}}{Z} \,\middle\|\, q_{\boldsymbol{\theta}}(\boldsymbol{x})\right)$$

“dissimilarity between two prob. dist.”

Training: Probability Density *Estimation ?*

Given a dataset, learn its probability density by minimizing the Negative Log-Likelihood

$$\mathrm{NLL}_{\boldsymbol{\theta}} = -\sum_{\boldsymbol{x}\in\mathrm{dataset}} \ln q_{\boldsymbol{\theta}}(\boldsymbol{x})$$

Network parameters

Equivalent to optimize the forward Kullback–Leibler divergence

$$\mathbb{KL}\left(\frac{e^{-E(\boldsymbol{x})}}{Z}\,\middle\|\, q_{\boldsymbol{\theta}}(\boldsymbol{x})\right)$$

“dissimilarity between two prob. dist.”

However, for typical stat-mech problems, we only have access to the bare energy function, not its samples

Training: Probability Density *Distillation*

Minimize the variational free energy

$$\mathcal{L}_{\boldsymbol{\theta}} = \int \mathrm{d}\boldsymbol{x}\, q_{\boldsymbol{\theta}}(\boldsymbol{x}) \left[\ln q_{\boldsymbol{\theta}}(\boldsymbol{x}) + E(\boldsymbol{x})\right]$$

Training: Probability Density *Distillation*

Minimize the variational free energy

$$\mathcal{L}_{\boldsymbol{\theta}} = \int \mathrm{d}\boldsymbol{x}\, q_{\boldsymbol{\theta}}(\boldsymbol{x}) \left[\ln q_{\boldsymbol{\theta}}(\boldsymbol{x}) + E(\boldsymbol{x})\right]$$

Energy function of the problem

Training: Probability Density *Distillation*

Minimize the variational free energy

$$\mathcal{L}_{\boldsymbol{\theta}} = \int \mathrm{d}\boldsymbol{x}\, q_{\boldsymbol{\theta}}(\boldsymbol{x}) \left[\ln q_{\boldsymbol{\theta}}(\boldsymbol{x}) + E(\boldsymbol{x})\right]$$

Entropy of model prob.

Energy function of the problem

Training: Probability Density *Distillation*

Minimize the variational free energy

$$\mathcal{L}_{\theta} = \int \mathrm{d}\boldsymbol{x}\, q_{\theta}(\boldsymbol{x}) \left[\ln q_{\theta}(\boldsymbol{x}) + E(\boldsymbol{x})\right]$$

"Learn from the samples generated by the network itself!"

Entropy of model prob.

Energy function of the problem

$$\mathcal{L}_{\theta} + \ln Z = \mathbb{KL}\left(q_{\theta}(\boldsymbol{x}) \,\middle\|\, \frac{e^{-E(\boldsymbol{x})}}{Z} \right) \geq 0$$

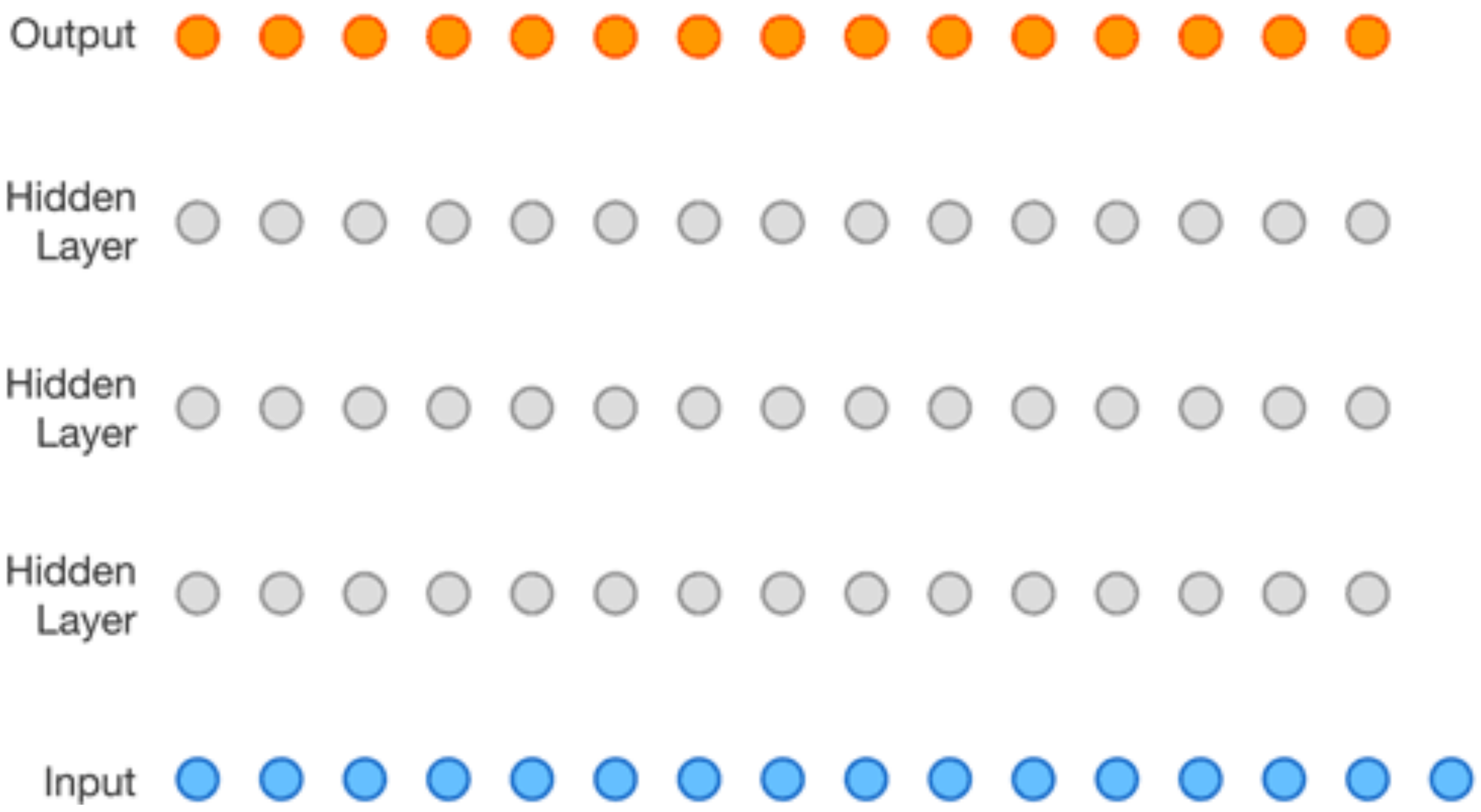
The loss function is lower bounded by the physical free energy (Gibbs-Bogoliubov-Feynman inequality)

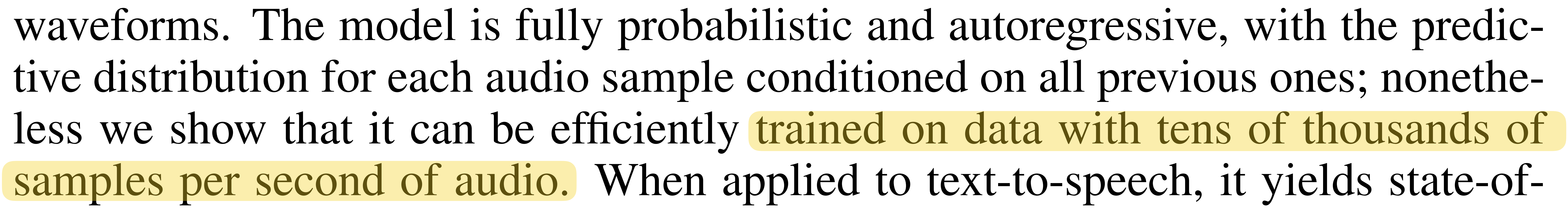
Interlude

https://www.youtube.com/watch?v=IXUQ-DdSDoE

Interlude: The WaveNet Story

WaveNet 2016
Autoregressive Flow

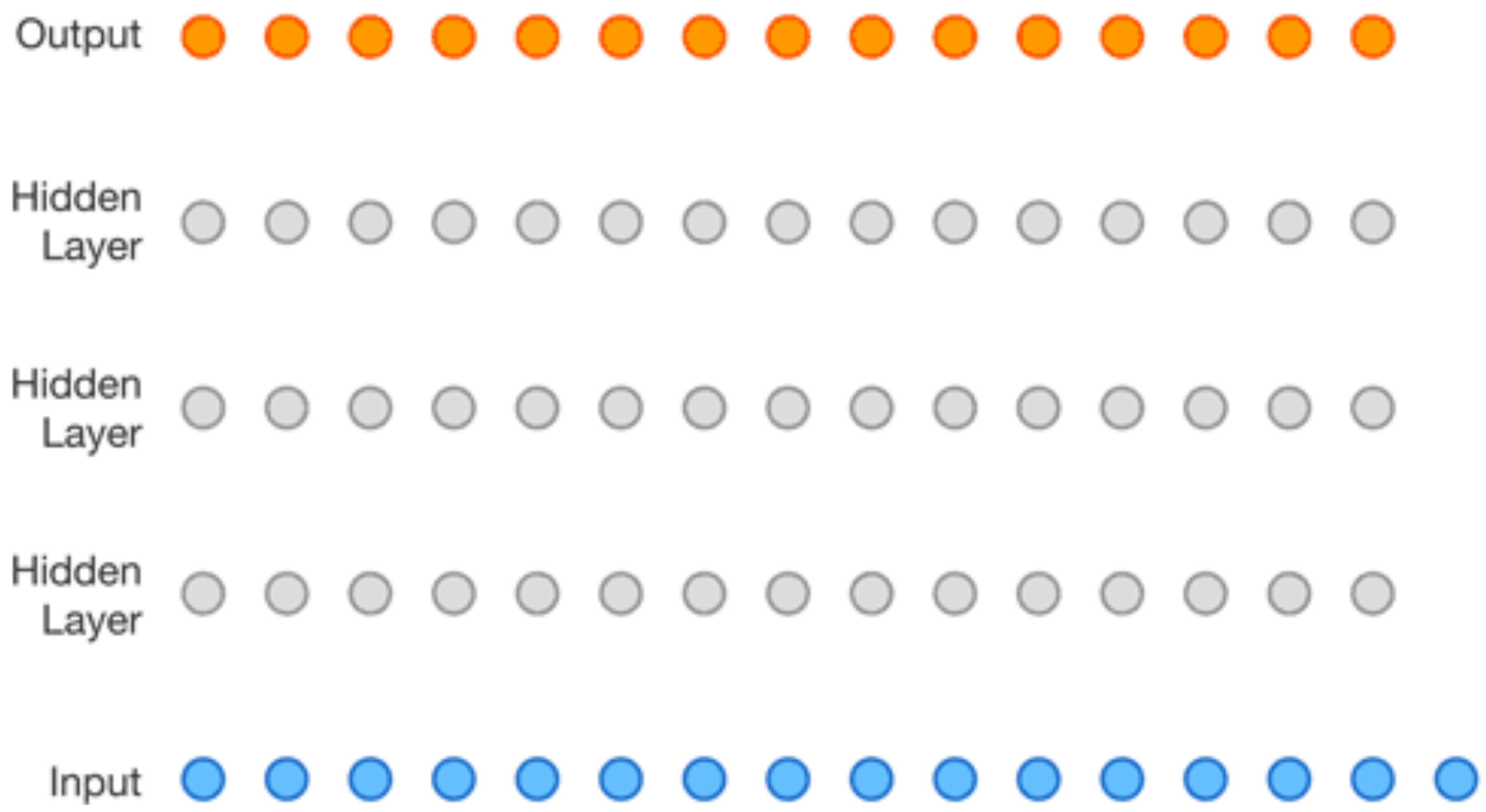


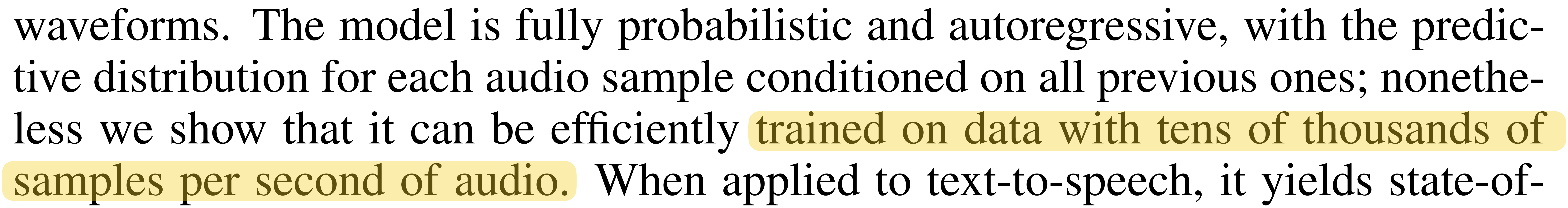

waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones; nonetheless we show that it can be efficiently trained on data with tens of thousands of samples per second of audio. When applied to text-to-speech, it yields state-of-

https://deepmind.com/blog/wavenet-generative-model-raw-audio/ 1609.03499
https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/ 1711.10433

Interlude: The WaveNet Story

WaveNet 2016
Autoregressive Flow




waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones; nonetheless we show that it can be efficiently trained on data with tens of thousands of samples per second of audio. When applied to text-to-speech, it yields state-of-

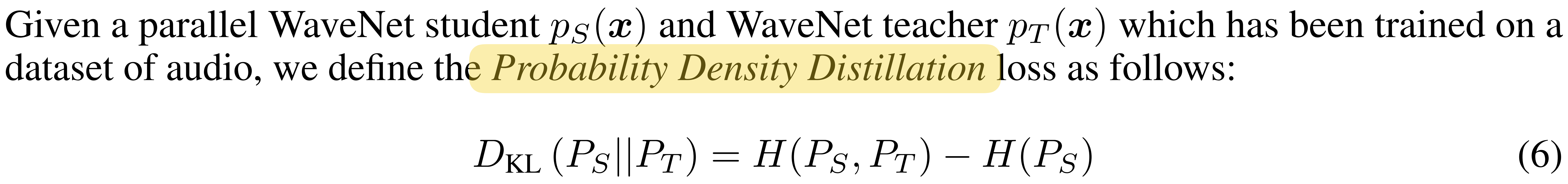
https://deepmind.com/blog/wavenet-generative-model-raw-audio/ 1609.03499
https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/ 1711.10433

Interlude: The WaveNet Story

Parallel WaveNet 2017
Inverse Autoregressive Flow

speech signal

input noise



Given a parallel WaveNet student $p_S(\boldsymbol{x})$ and WaveNet teacher $p_T(\boldsymbol{x})$ which has been trained on a dataset of audio, we define the *Probability Density Distillation* loss as follows:

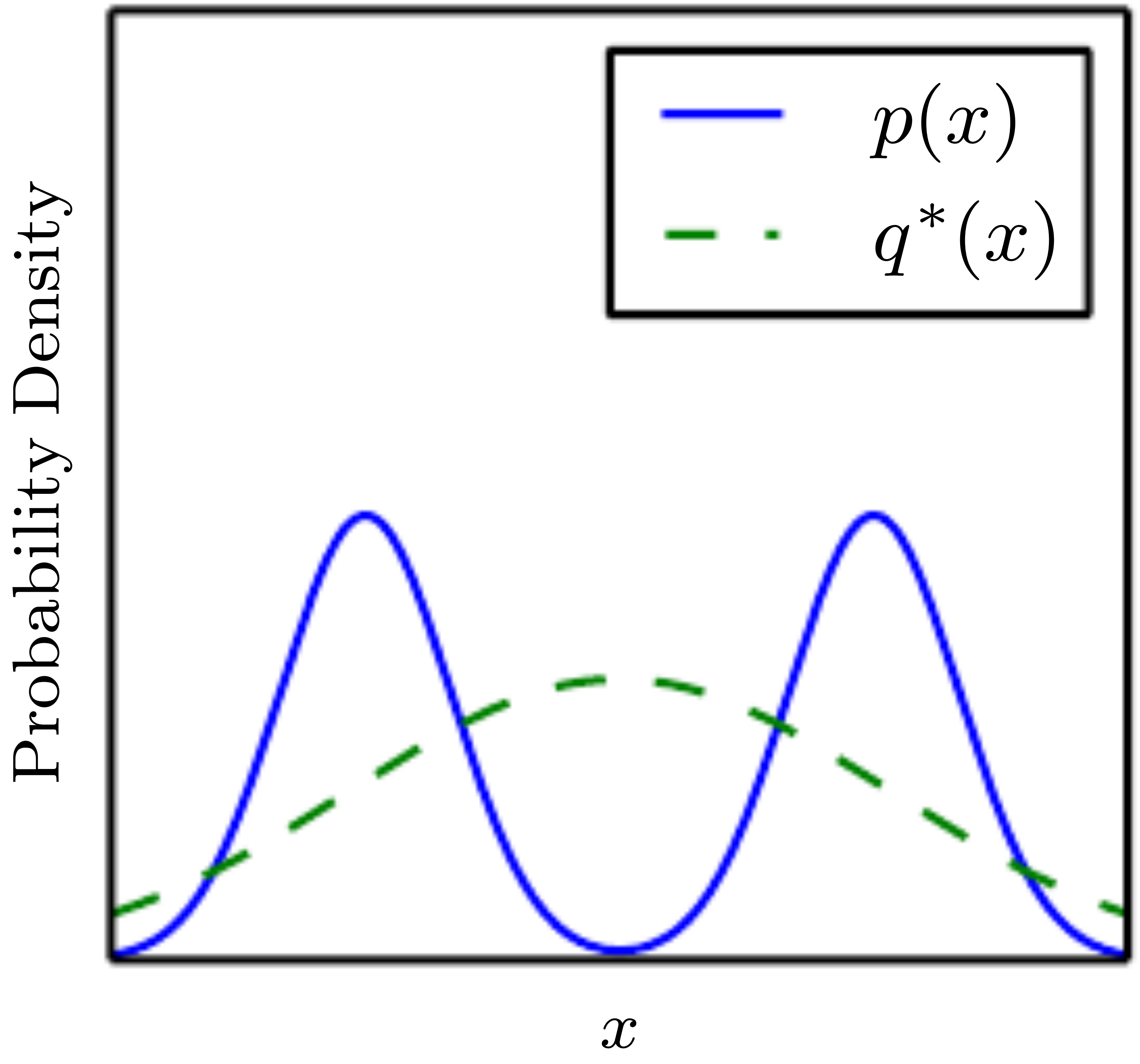
$$D_{\text{KL}}\left(P_S \| P_T\right) = H(P_S, P_T) - H(P_S) \tag{6}$$

https://deepmind.com/blog/wavenet-generative-model-raw-audio/ 1609.03499
https://deepmind.com/blog/high-fidelity-speech-synthesis-wavenet/ 1711.10433

Forward KL or Reverse KL ?

Maximum Likelihood Estimation

$$q^* = \text{argmin}_q D_{\text{KL}}(p\|q)$$



Probability Density Distillation

$$q^* = \text{argmin}_q D_{\text{KL}}(q\|p)$$

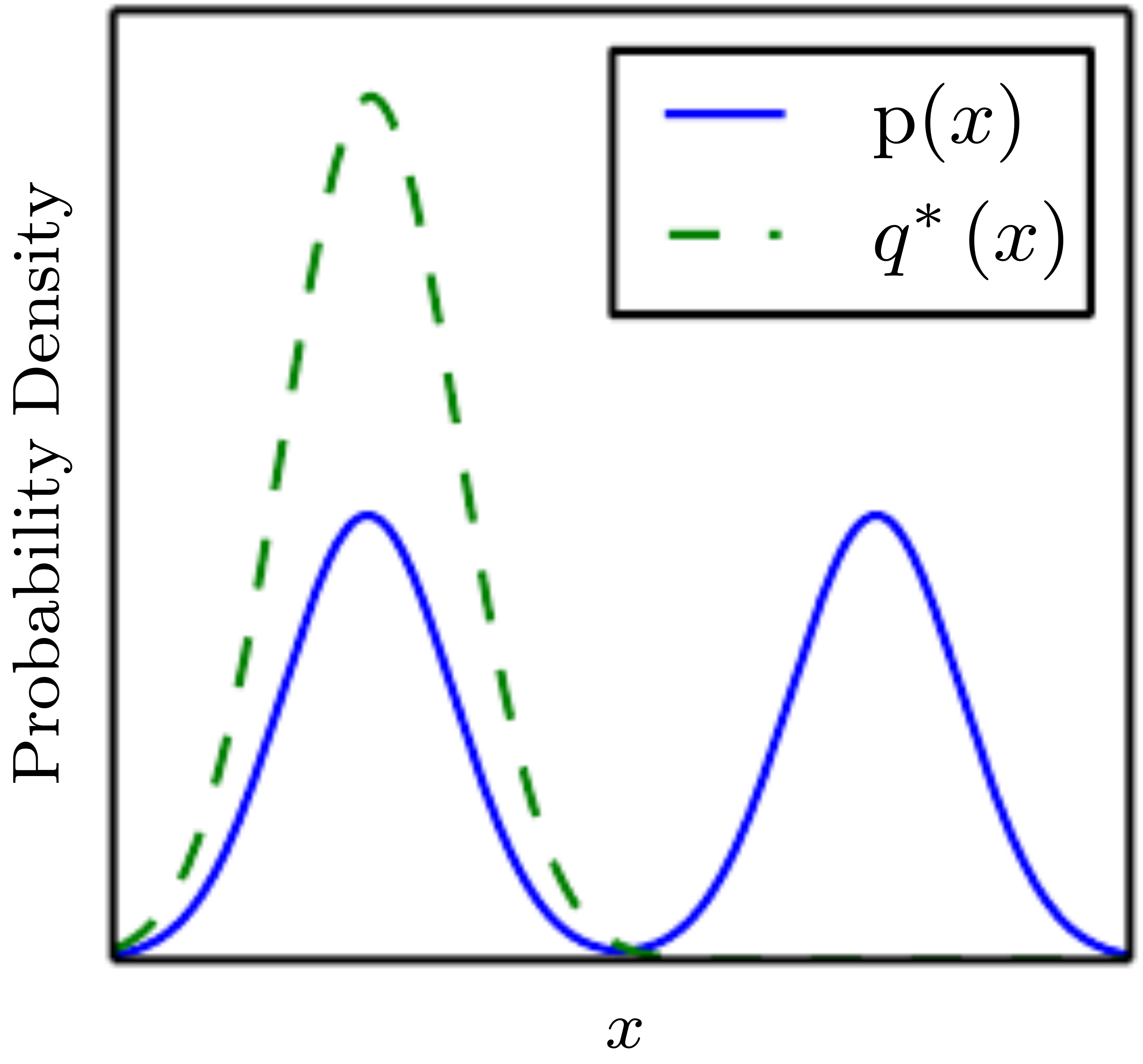


Fig. 3.6, Goodfellow, Bengio, Courville, http://www.deeplearningbook.org/

“Reparametrization trick”

Unbiased, low variance gradient estimator w.r.t. random sampling

$$\mathcal{L}_\theta = \mathop{\mathbb{E}}_{z \sim p(z)} \left[\ln q(g_\theta(z)) + E(g_\theta(z)) \right]$$

Sample from the prior dist.

Network parameters

Secret behind scalable deep learning:
end-to-end training via **back-propagation**

“Reparametrization trick”

Unbiased, low variance gradient estimator w.r.t. random sampling

$$\mathcal{L}_\theta = \mathop{\mathbb{E}}_{z\sim p(z)} \left[\ln q(g_\theta(z)) + E(g_\theta(z))\right]$$

Sample from the prior dist.

Network parameters

1. Draw z from prior
2. Pass them through the network x=g(z)
3. Evaluate the variational loss
4. Optimize

Secret behind scalable deep learning:
end-to-end training via **back-propagation**

Let’s apply it to the Ising model!

$$\pi(\boldsymbol{s}) = \exp\left(\frac{1}{2}\boldsymbol{s}^T K \boldsymbol{s}\right)$$

Let’s apply it to the Ising model!

$$\pi(\boldsymbol{s}) = \exp\left(\frac{1}{2}\boldsymbol{s}^T K \boldsymbol{s}\right)$$

decouple

M. E. Fisher 1983

Binney et al 1992

$$\propto \int \mathrm{d}\boldsymbol{x} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \left(K + \alpha I\right)^{-1} \boldsymbol{x} + \boldsymbol{s}^T \boldsymbol{x}\right)$$

Let's apply it to the Ising model!

$$\pi(\boldsymbol{s}) = \exp\left(\frac{1}{2}\boldsymbol{s}^T K \boldsymbol{s}\right)$$

decouple

M. E. Fisher 1983

Binney et al 1992

$$\propto \int \mathrm{d}\boldsymbol{x} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \left(K + \alpha I\right)^{-1} \boldsymbol{x} + \boldsymbol{s}^T \boldsymbol{x}\right)$$

trace out s

$$\boxed{\pi(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\boldsymbol{x}^T \left(K + \alpha I\right)^{-1} \boldsymbol{x}\right) \prod_i \cosh(x_i)}$$

Let's apply it to the Ising model!

$$\pi(\boldsymbol{s}) = \exp\left(\frac{1}{2}\boldsymbol{s}^T K \boldsymbol{s}\right)$$

decouple

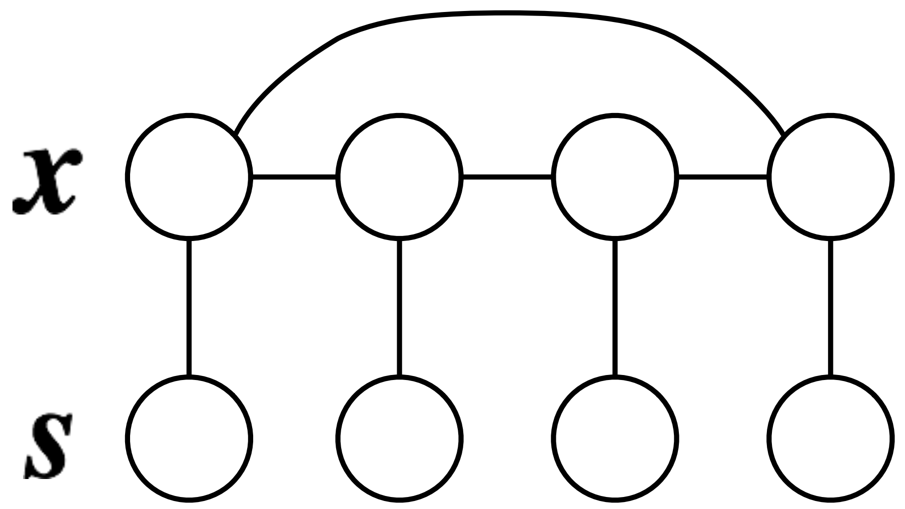
M. E. Fisher 1983

Binney et al 1992

$$\propto \int \mathrm{d}\boldsymbol{x} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \left(K + \alpha I\right)^{-1} \boldsymbol{x} + \boldsymbol{s}^T \boldsymbol{x}\right)$$

trace out s

$$\boxed{\pi(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\boldsymbol{x}^T \left(K + \alpha I\right)^{-1} \boldsymbol{x}\right) \prod_i \cosh(x_i)}$$



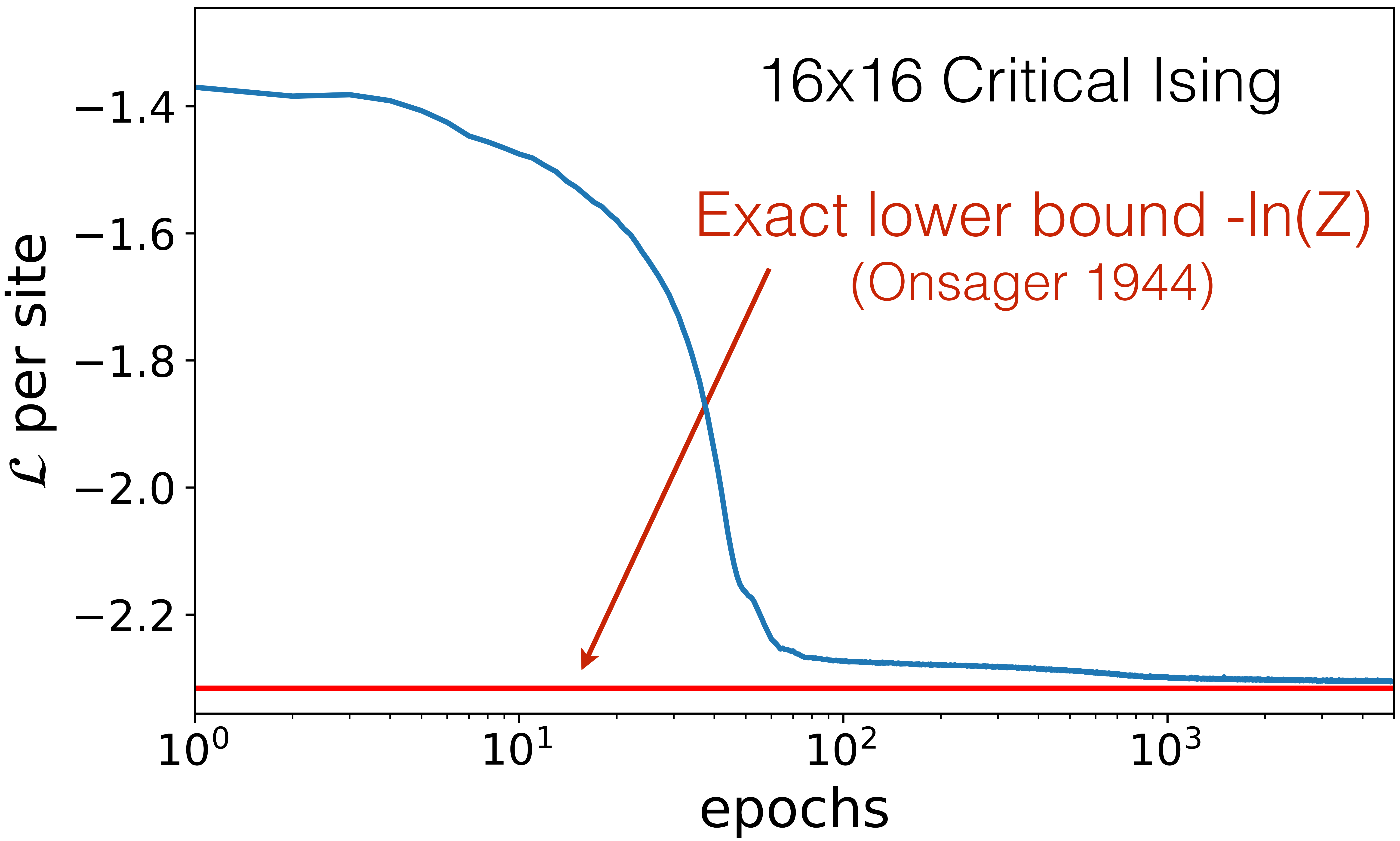
$$\pi(\boldsymbol{s}|\boldsymbol{x}) = \prod_i \left(1 + e^{-2 s_i x_i}\right)^{-1}$$

continuous dual of the Ising model

"Gaussian-Bernoulli Boltzmann Machine"

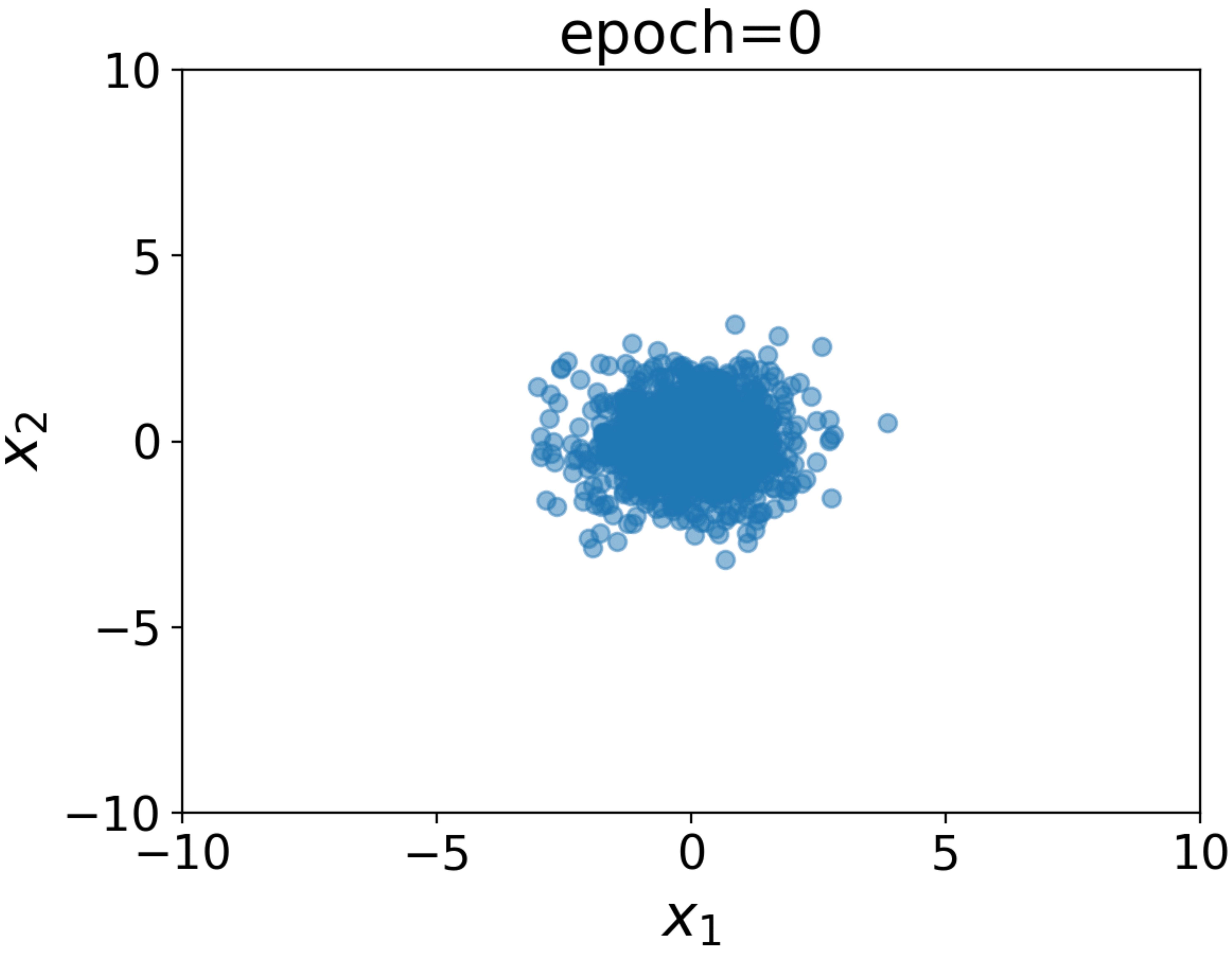
Zhang, Sutton, Storkey, Ghahramani, NIPS 2012

Variational Loss

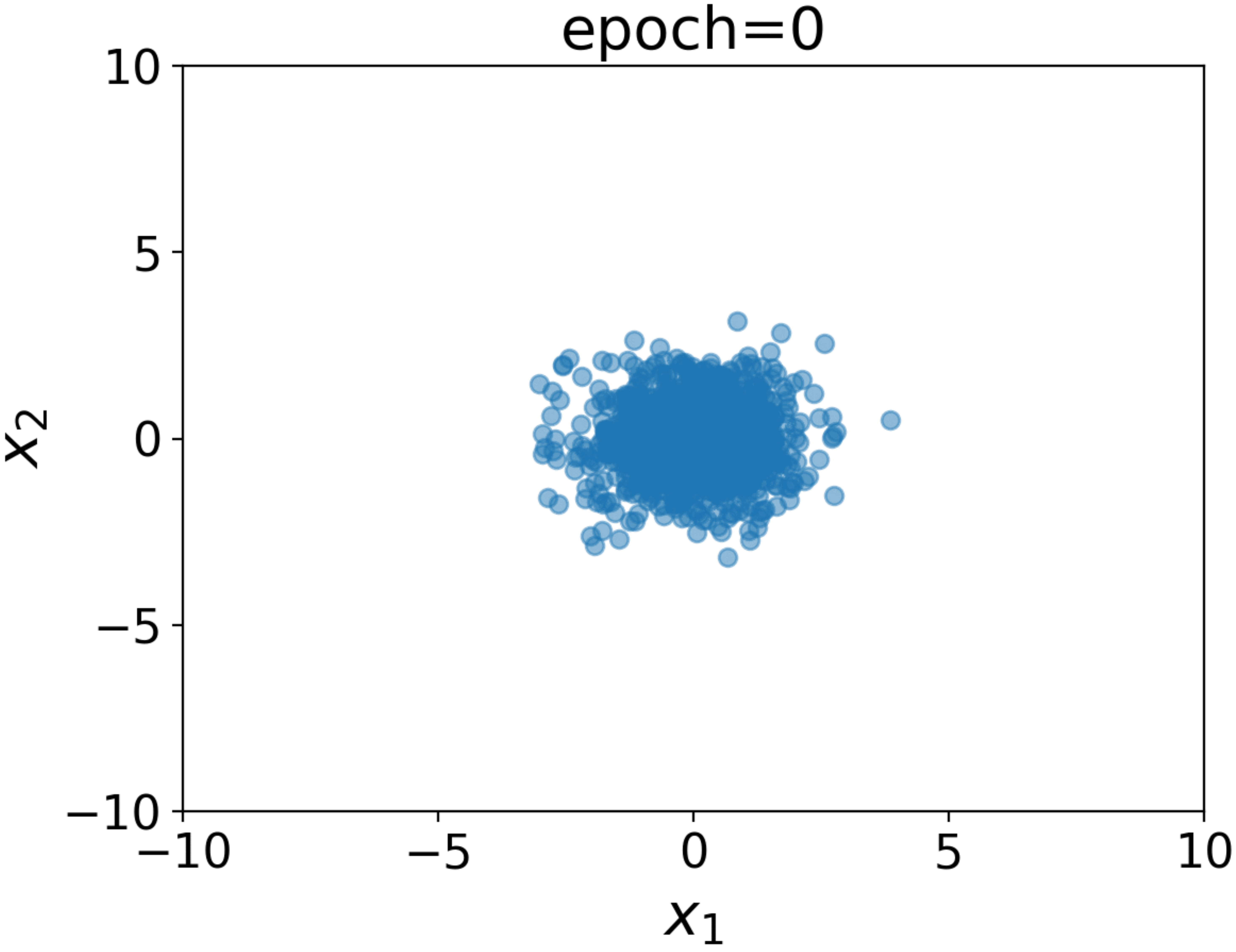


Training = Variational free energy calculation

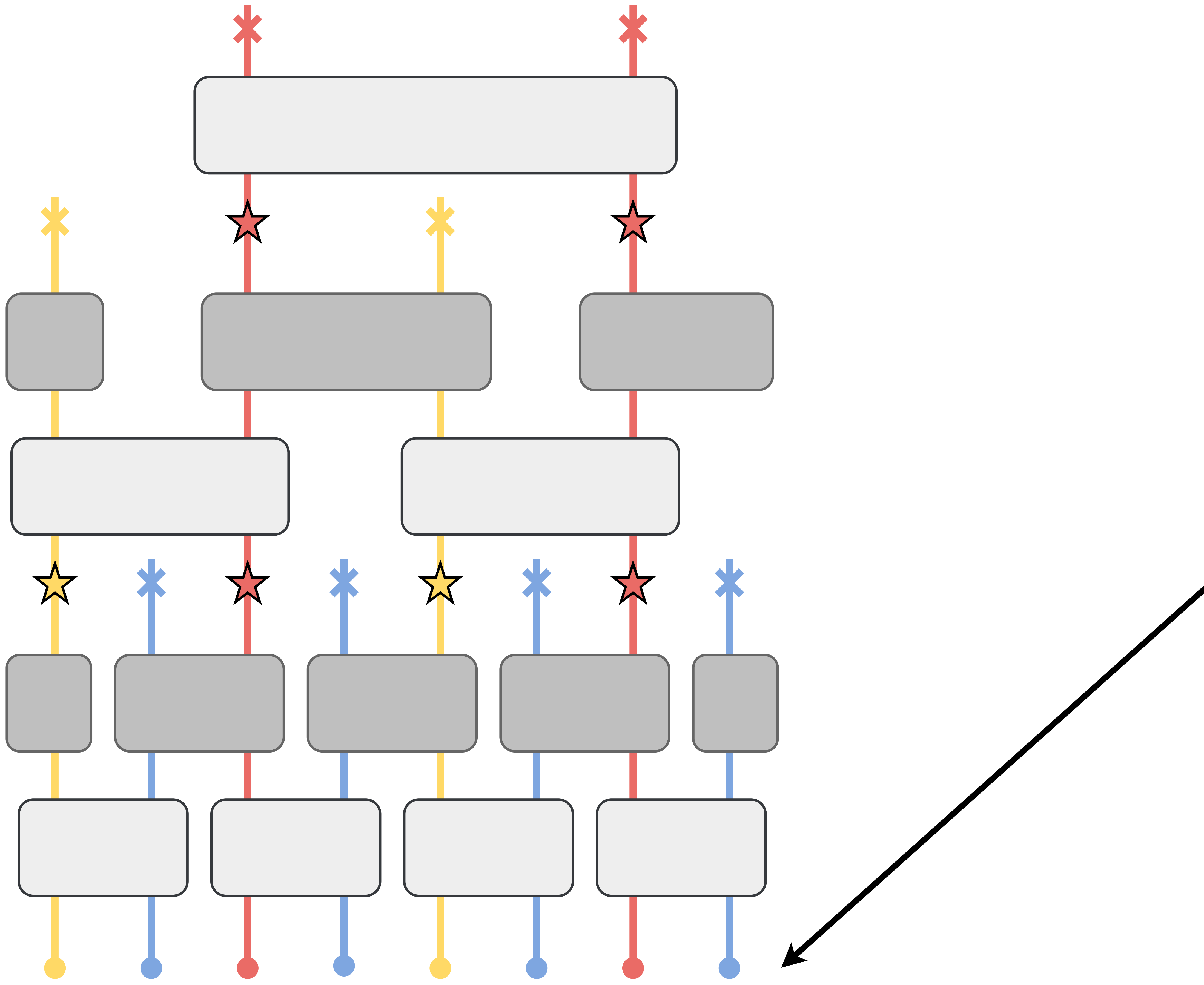
Generated Samples



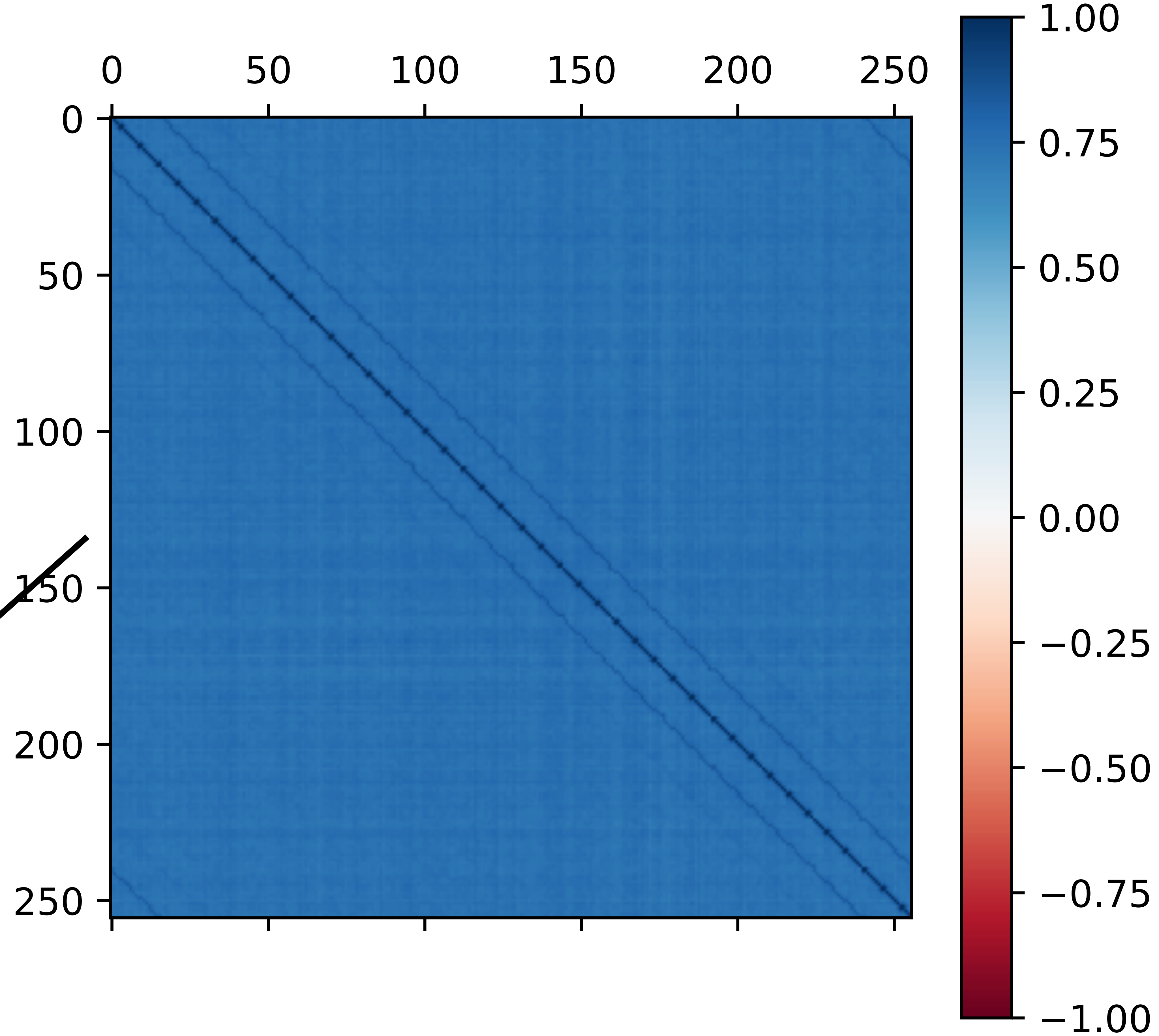
Generated Samples



What is the neural net doing?



Physical variables

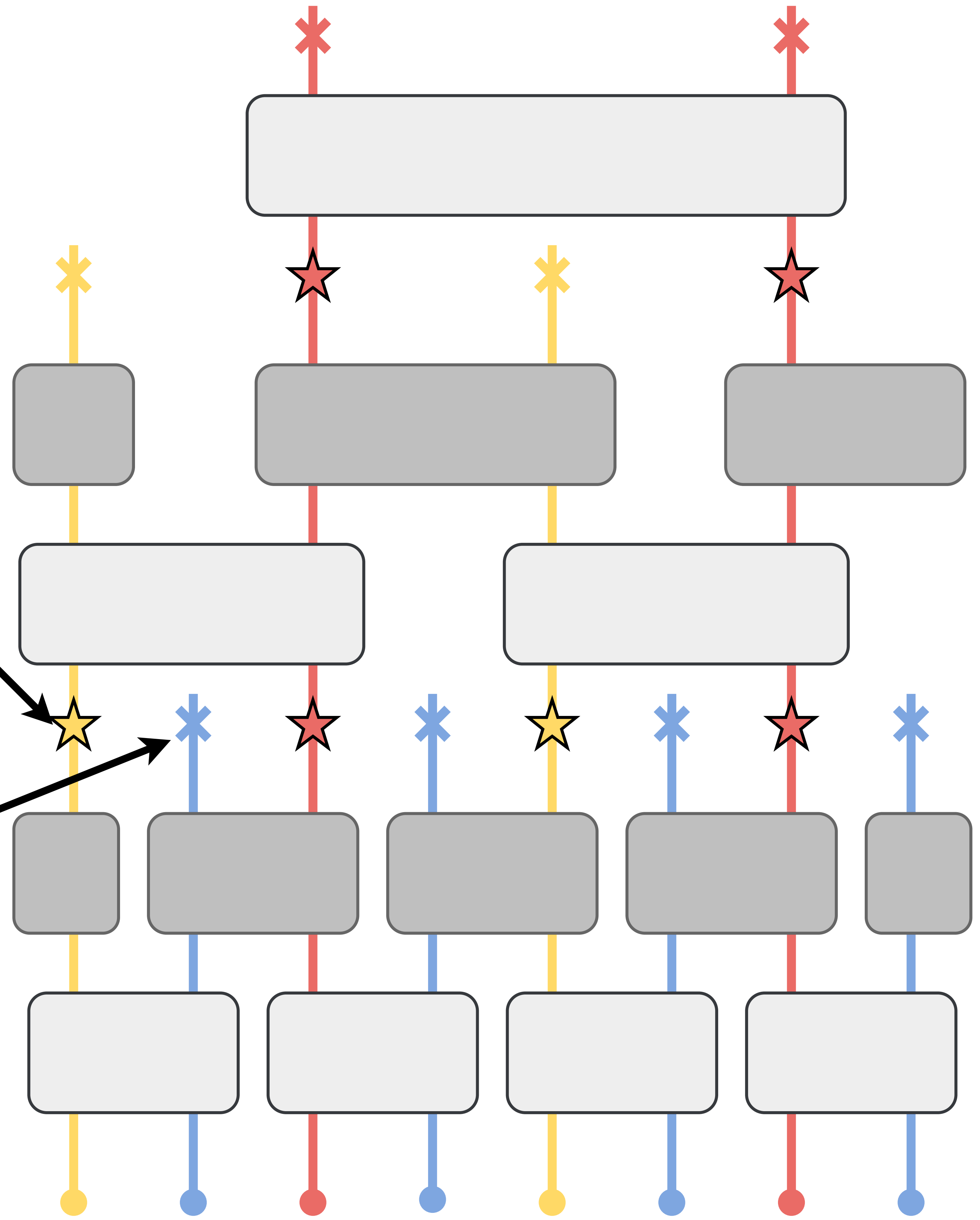


Two-point correlations

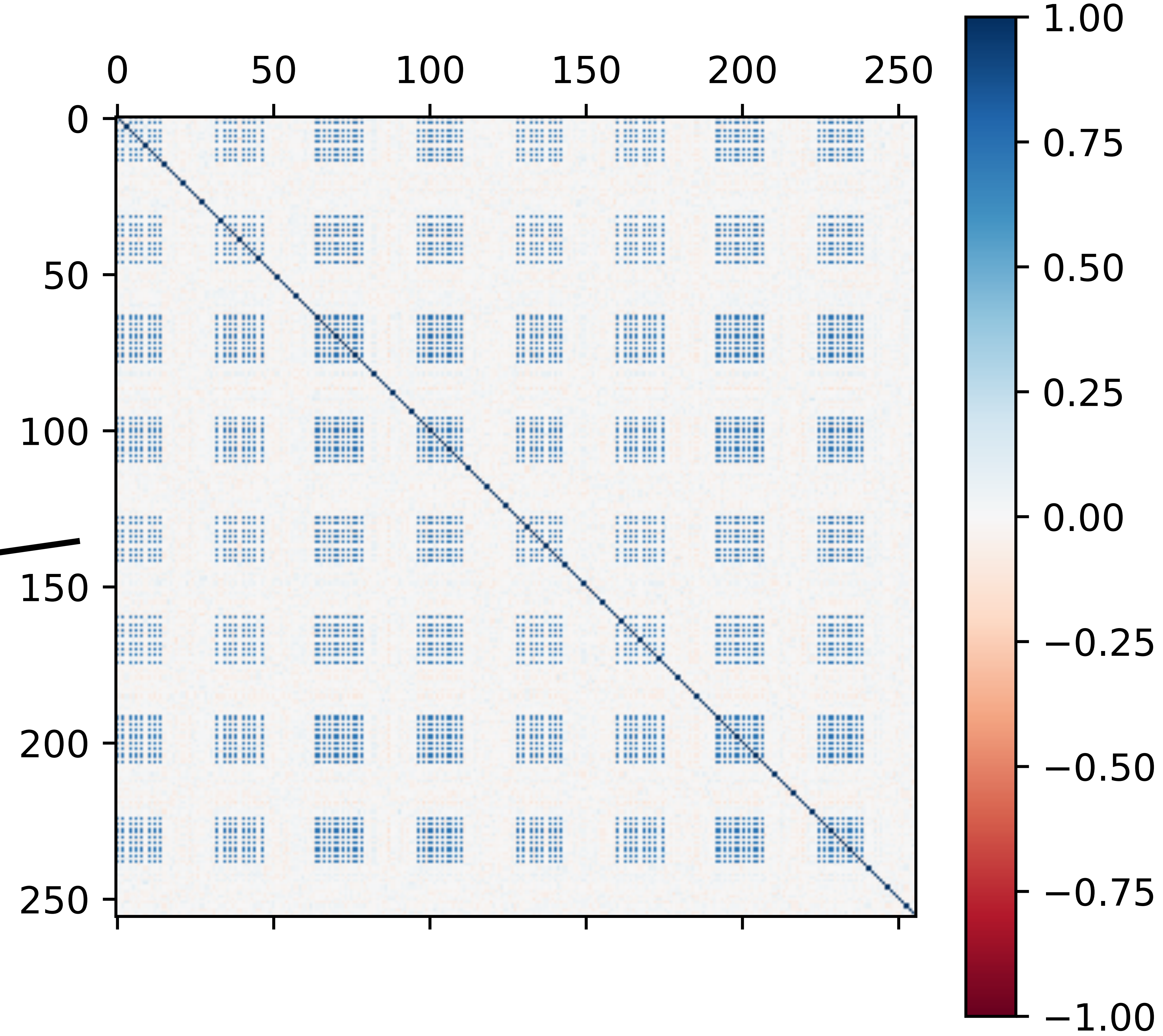
What is the neural net doing?

Collective variables

Latent variables

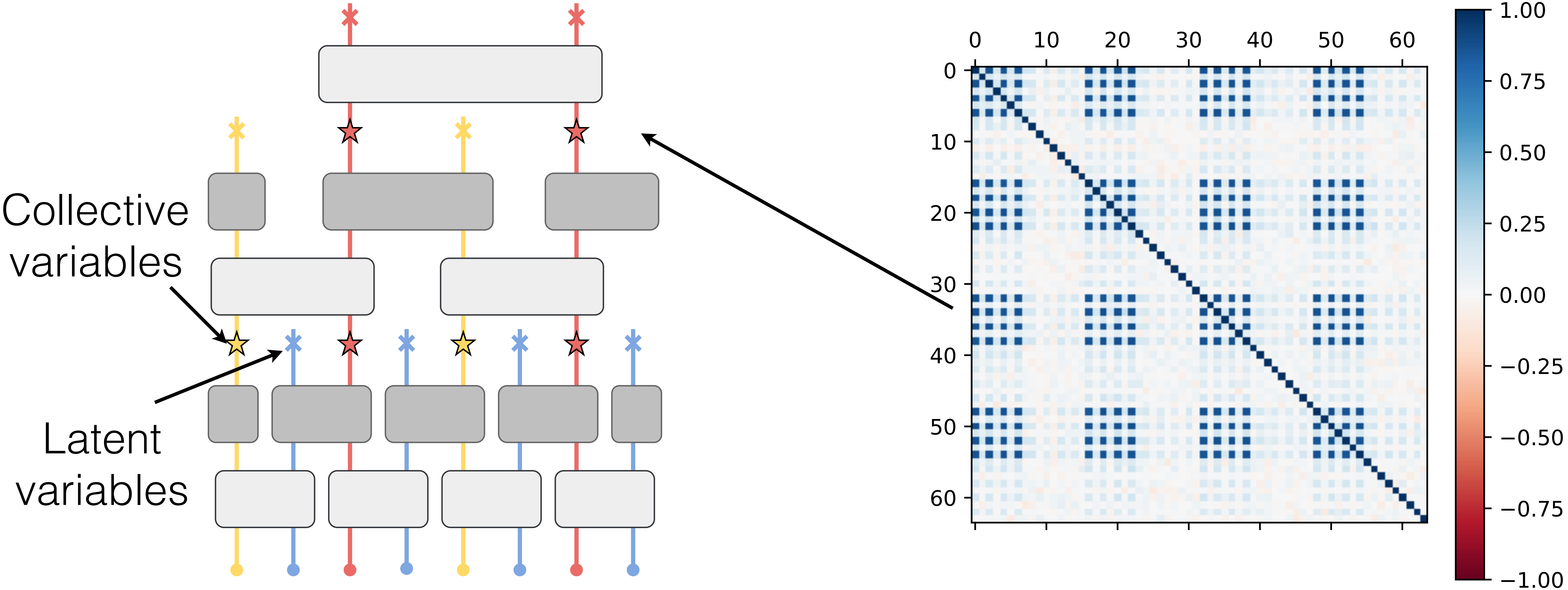


Physical variables



Two-point correlations

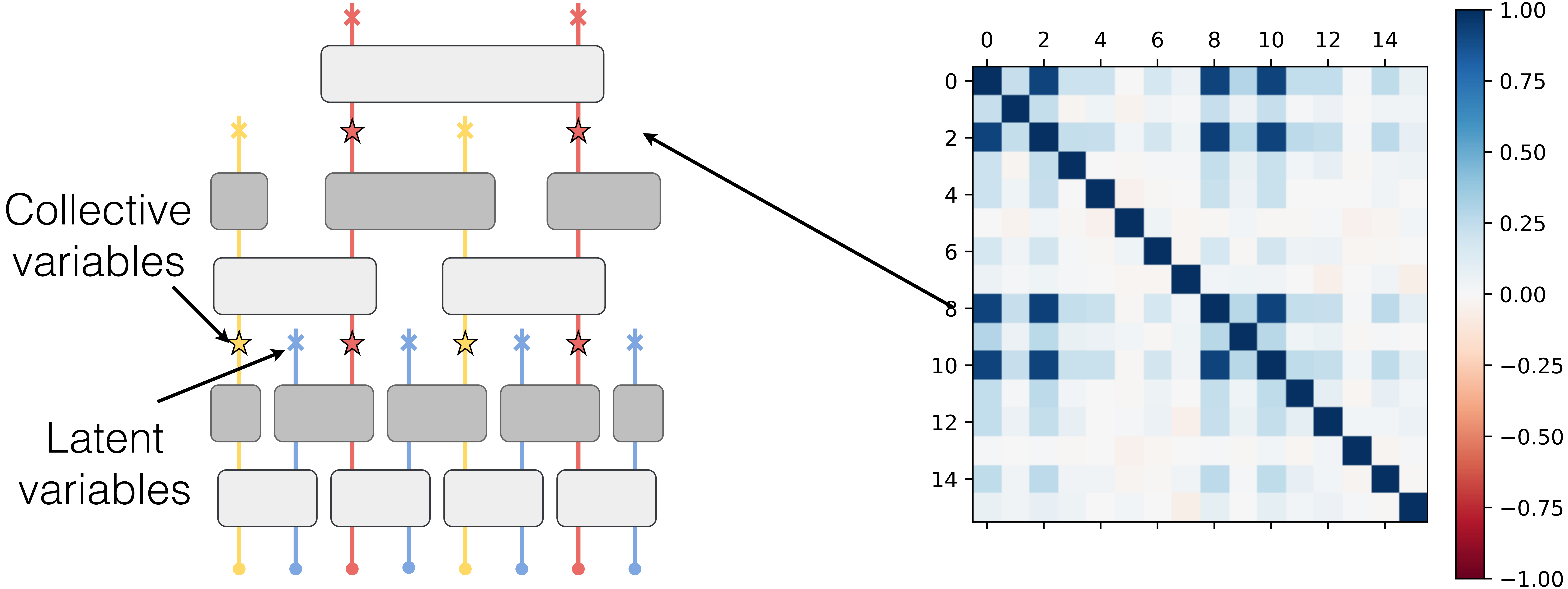
What is the neural net doing?



Physical variables

Two-point correlations

What is the neural net doing?



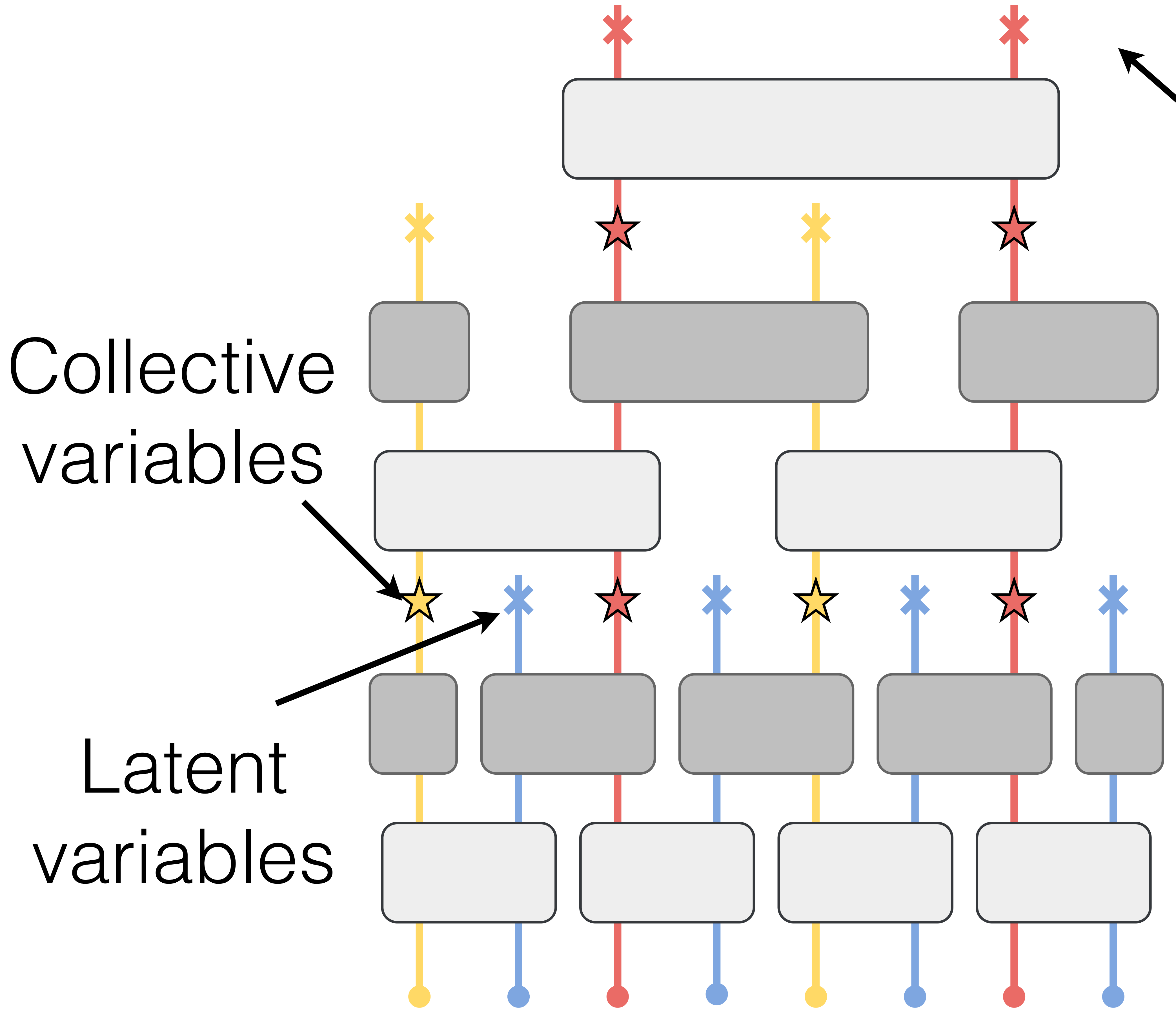
Collective variables

Latent variables

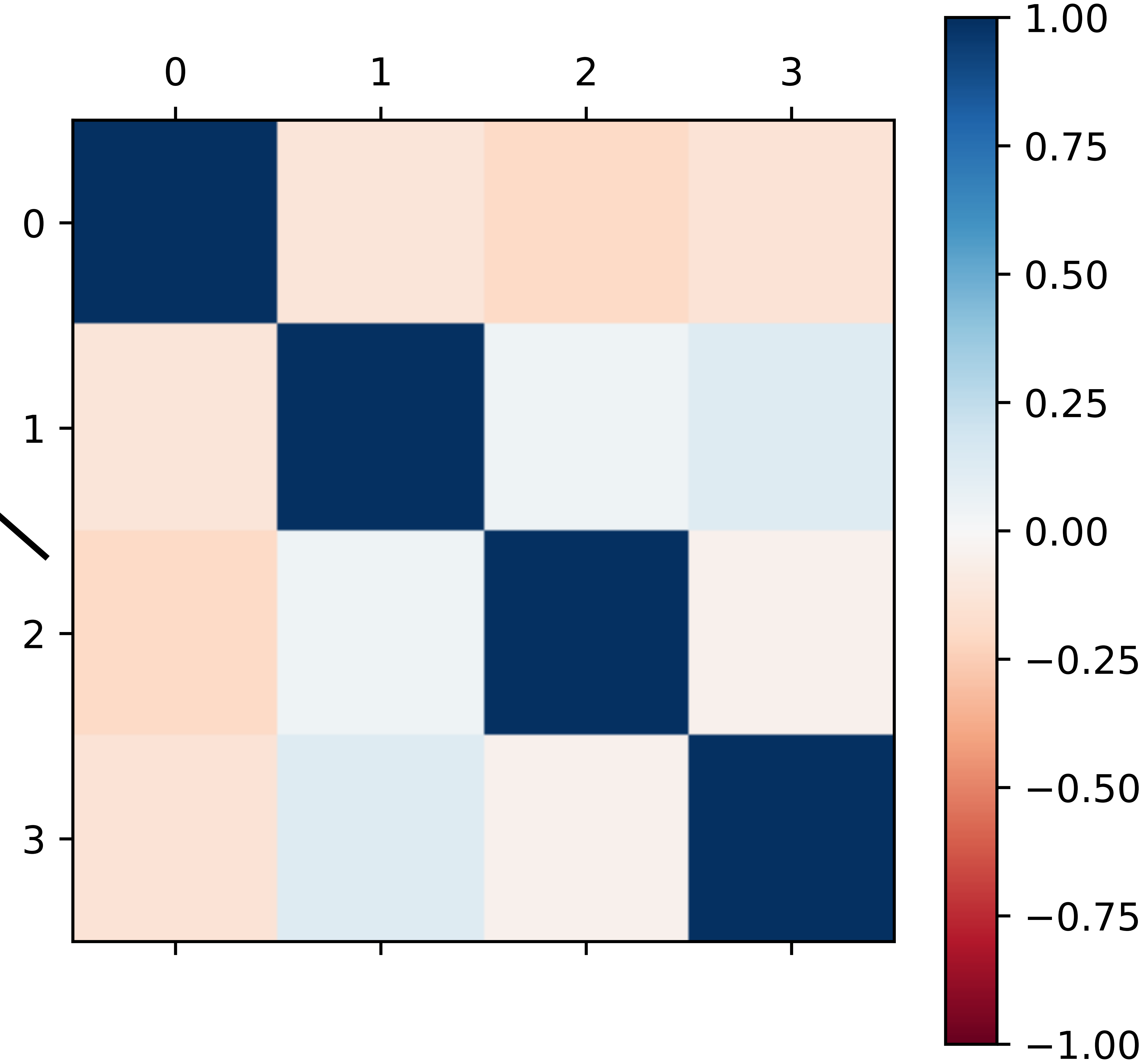
Physical variables

Two-point correlations

What is the neural net doing?



Physical variables



Two-point correlations

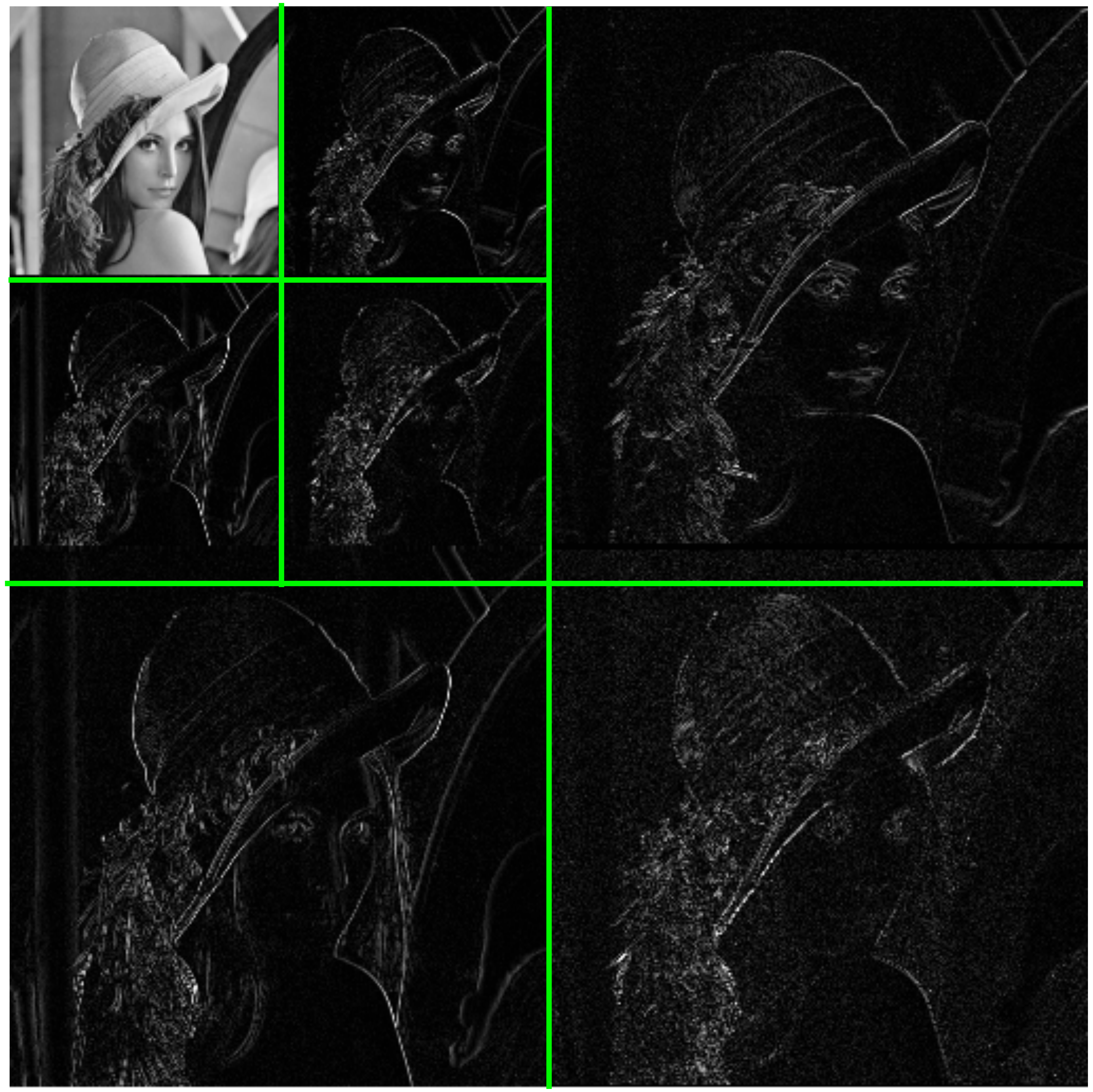
How to interpret the latent variables ?

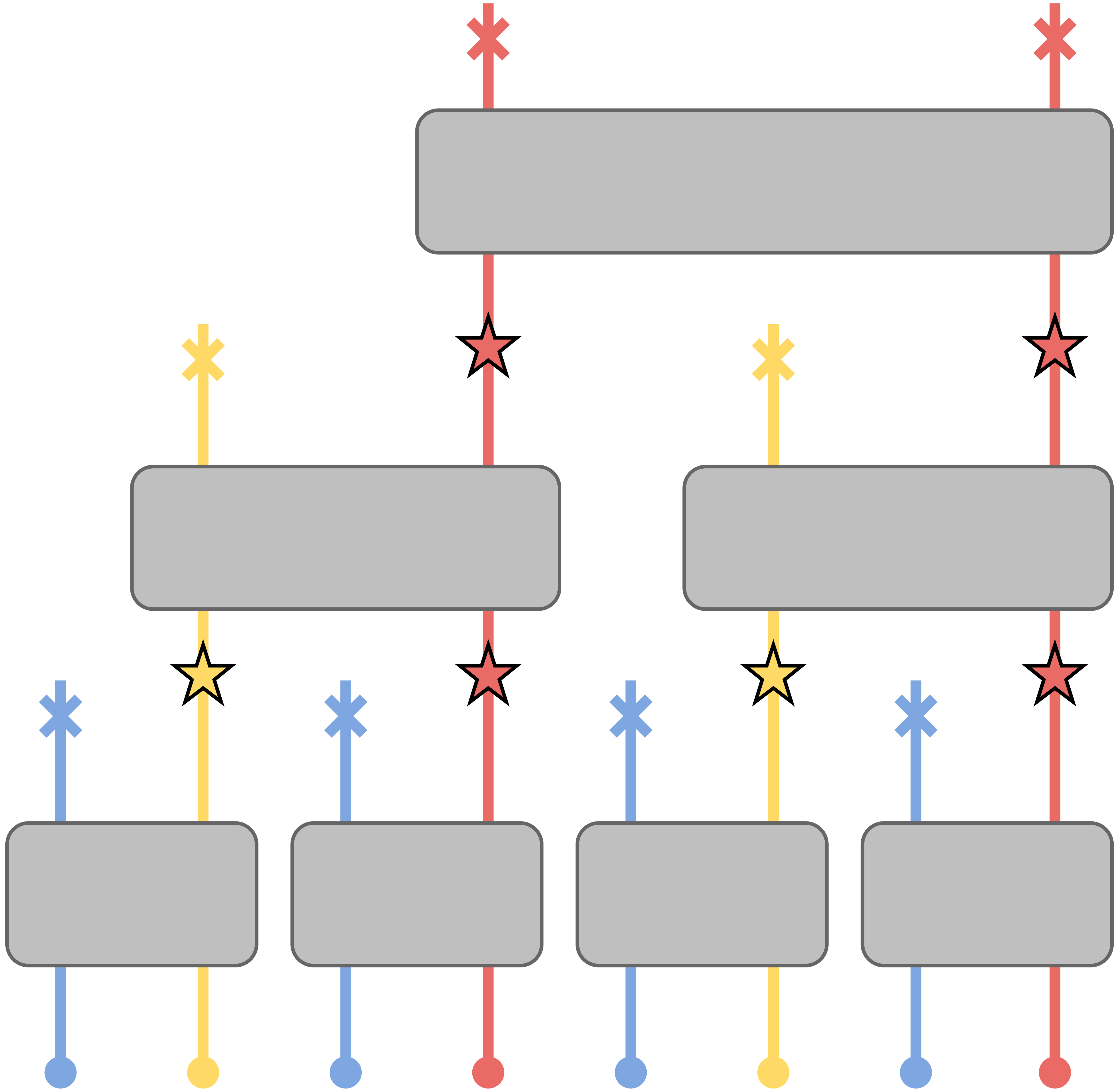
How to interpret the latent variables?

Guy, Wavelets & RG, 1999+

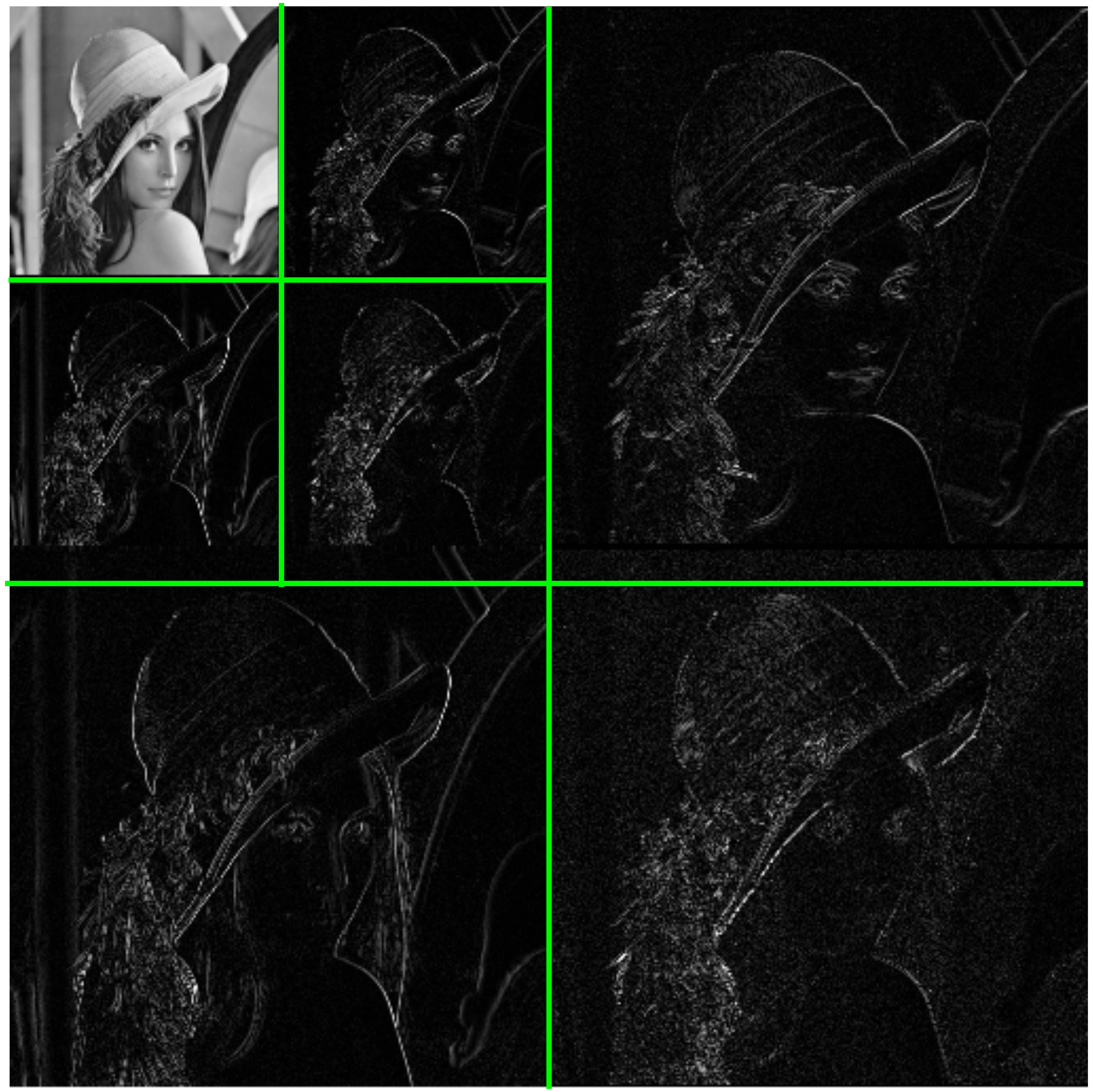
White, Evenbly, Qi, Wavelets, MERA, and holographic mapping 2013+

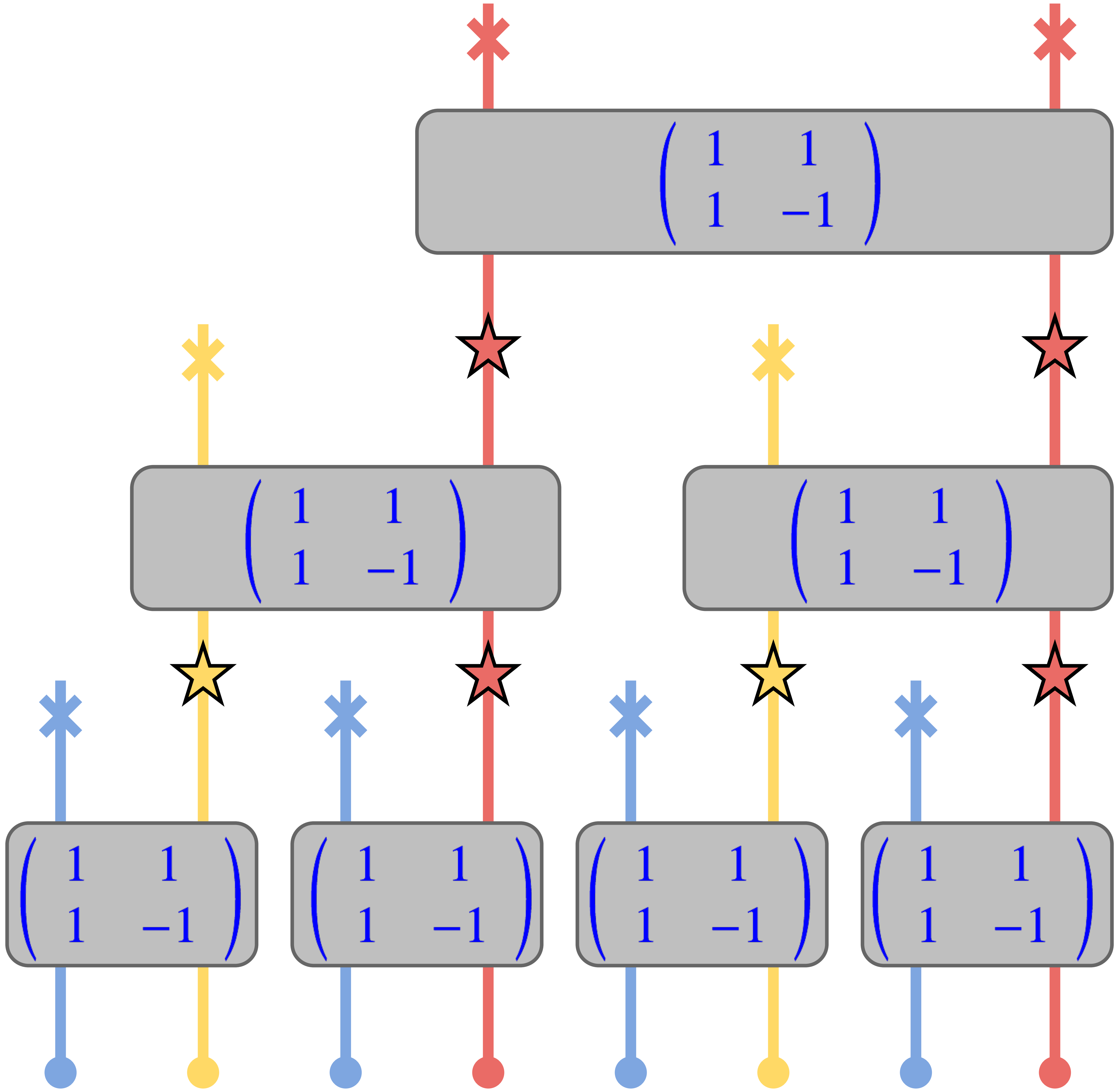
Wavelet transformation for Lena and Ising





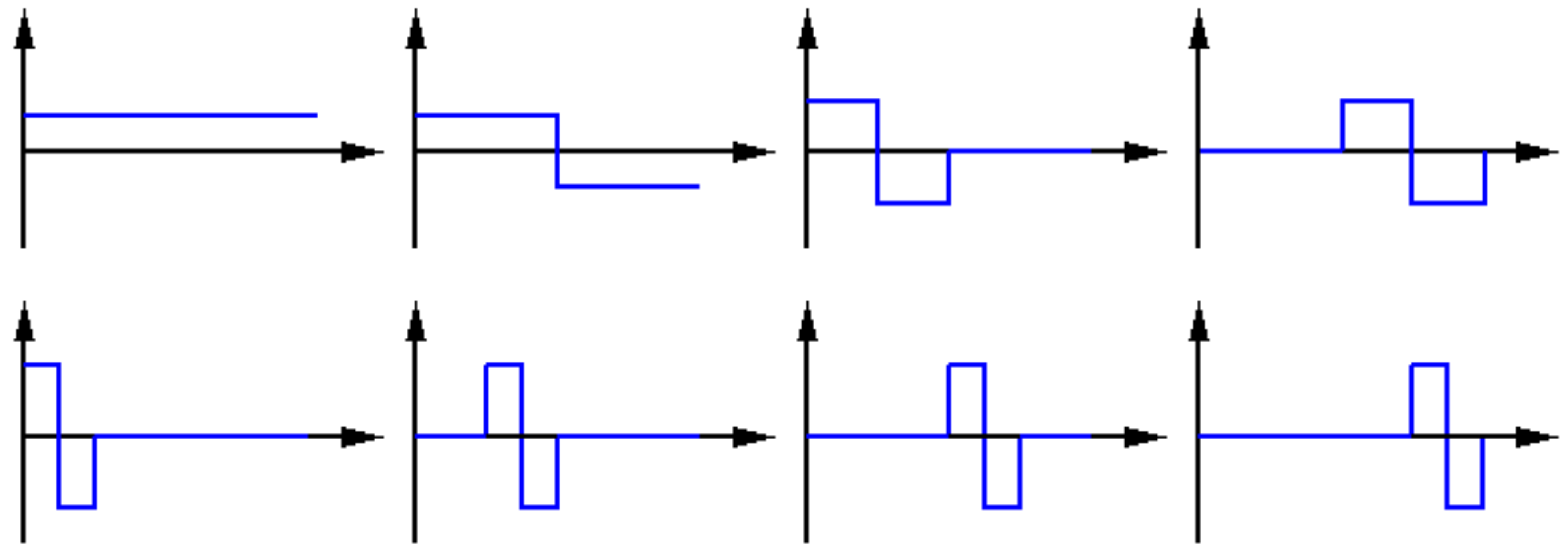
Wavelet transformation for Lena and Ising

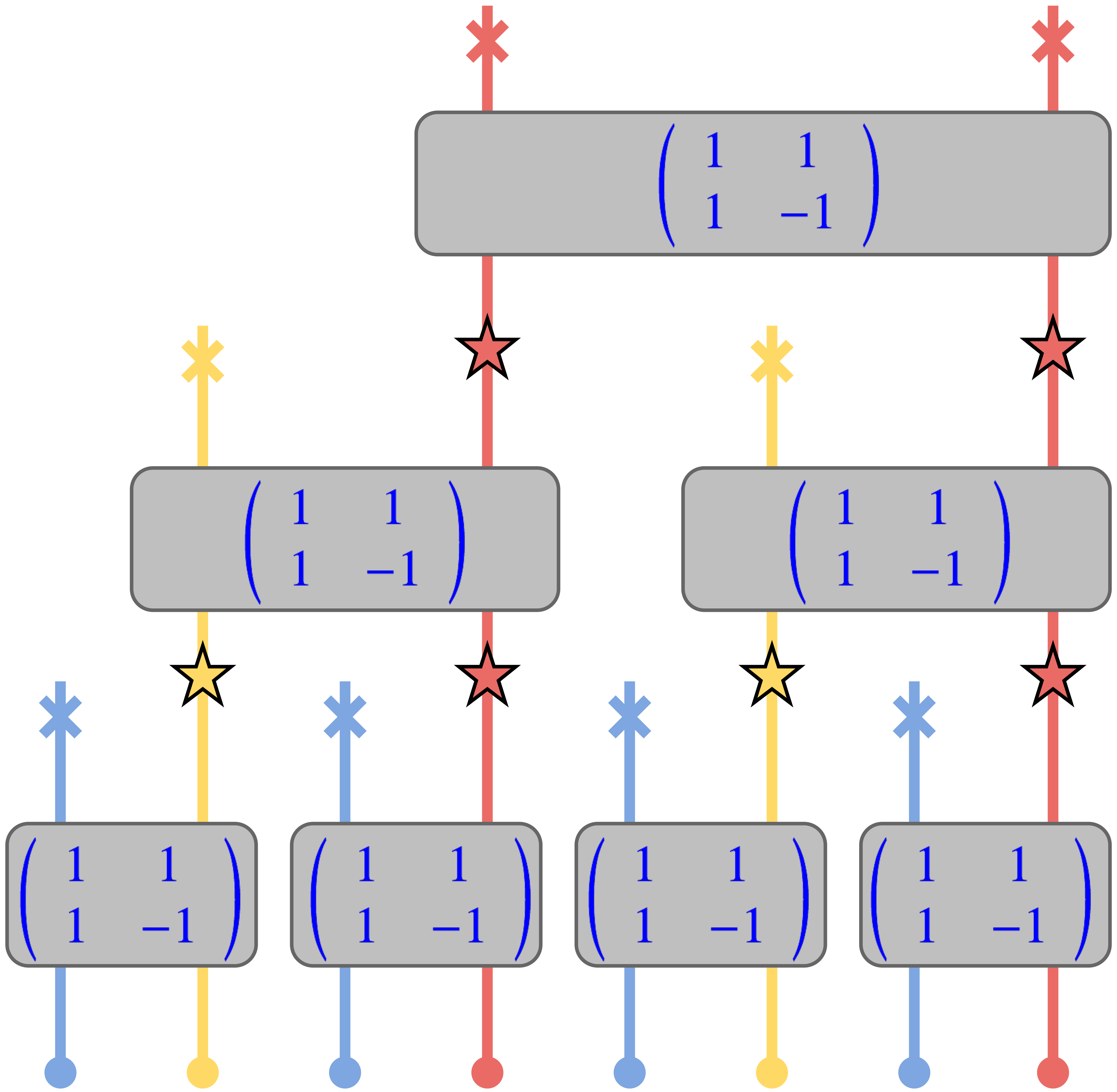


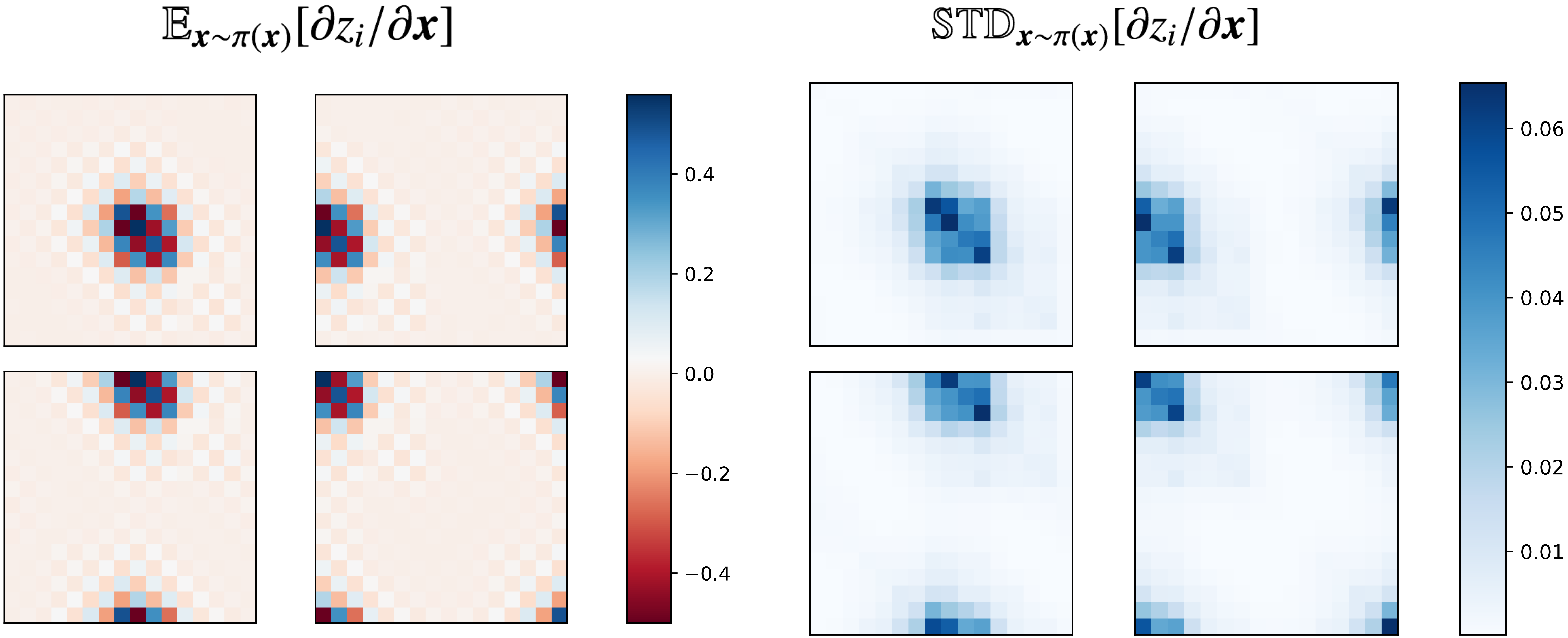

$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

Wavelet transformation for Lena and Ising

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$





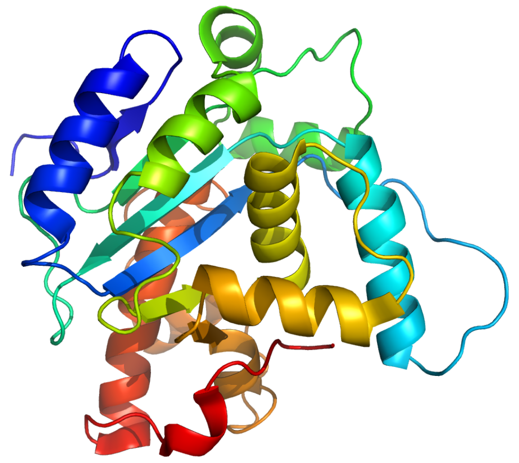


$\mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x})}[\partial z_i/\partial \boldsymbol{x}]$

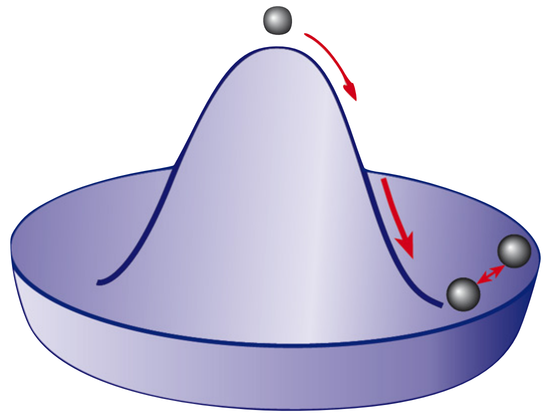
$\mathbb{STD}_{\boldsymbol{x}\sim\pi(\boldsymbol{x})}[\partial z_i/\partial \boldsymbol{x}]$

The latent variables seem to be nonlinear & adaptive generalizations of wavelets

How is this useful ?

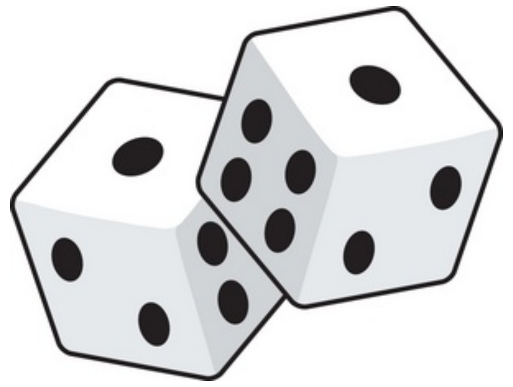


Identifying mutually independent collective variables (molecular simulation, PIMC, PIMD)



Deriving effective field theory of collective variables

Information preserving RG for holographic mapping



Accelerated Monte Carlo simulation

A Comparison of two Markov Chain Monte Carlo samplers

How to transform *almost* anything to a Gaussian ?

Normalizing flow

$$Z = \int \mathrm{d}\boldsymbol{x}\, \boxed{\pi(\boldsymbol{x})} = \int \mathrm{d}z\, \pi(g(z)) \left|\det\left(\frac{\partial g(z)}{\partial z}\right)\right| = \int \mathrm{d}z\, p(z) \left[\frac{\pi(g(z))}{q(g(z))}\right]$$

Physical
Prob. Dist.

**Learnable change-of-variables for
a mutually independent representation**

How to transform *almost* anything to a Gaussian ?

Normalizing flow

$$Z = \int \mathrm{d}\boldsymbol{x}\, \boxed{\pi(\boldsymbol{x})} = \int \mathrm{d}z\, \boxed{\pi(g(z)) \left|\det\left(\frac{\partial g(z)}{\partial z}\right)\right|} = \int \mathrm{d}z\, p(z) \left[\frac{\pi(g(z))}{q(g(z))}\right]$$

Physical
Prob. Dist.

Latent space
Prob. Dist.

Learnable change-of-variables for
a mutually independent representation

How to transform *almost* anything to a Gaussian ?

Normalizing flow

$$Z = \int \mathrm{d}\boldsymbol{x}\, \boxed{\pi(\boldsymbol{x})} = \int \mathrm{d}z\, \boxed{\pi(g(z)) \left|\det\left(\frac{\partial g(z)}{\partial z}\right)\right|} = \int \mathrm{d}z\, \boxed{p(z)} \left[\frac{\pi(g(z))}{q(g(z))}\right]$$

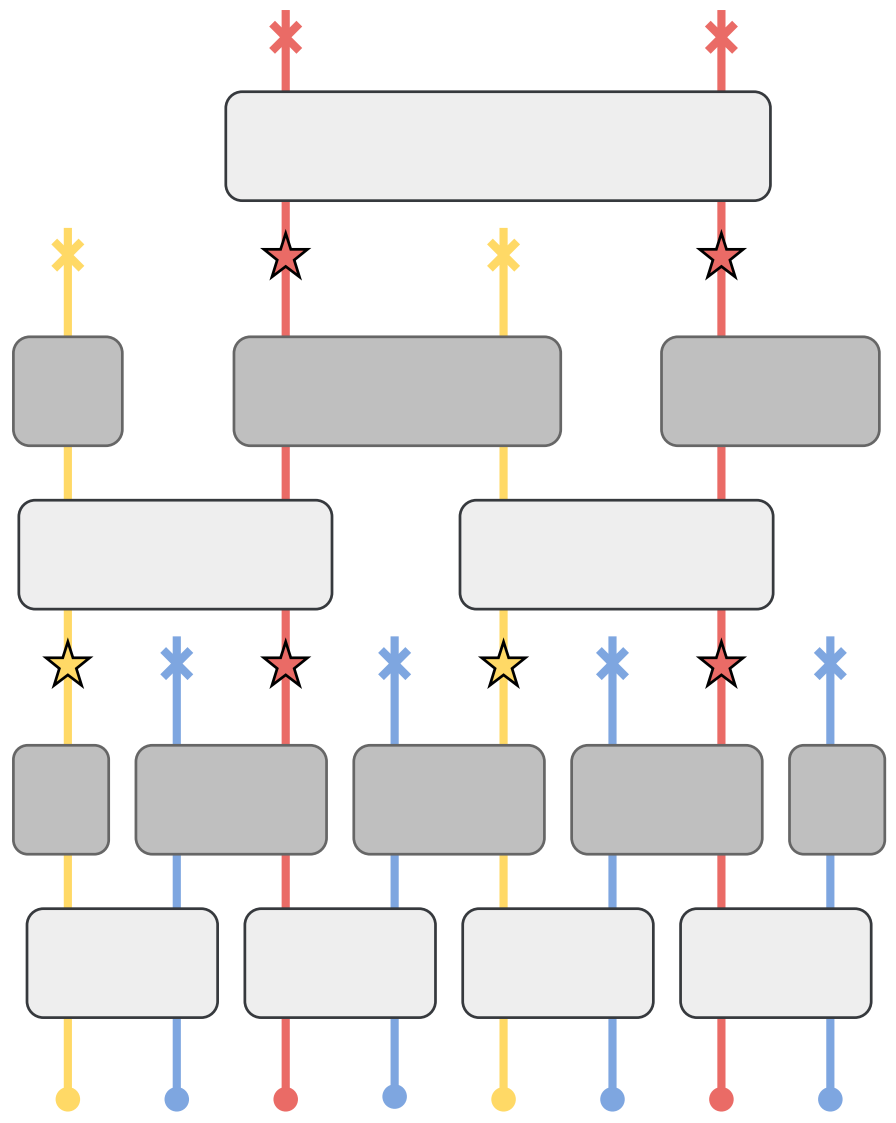
Physical
Prob. Dist.

Latent space
Prob. Dist.

Prior. Dist.

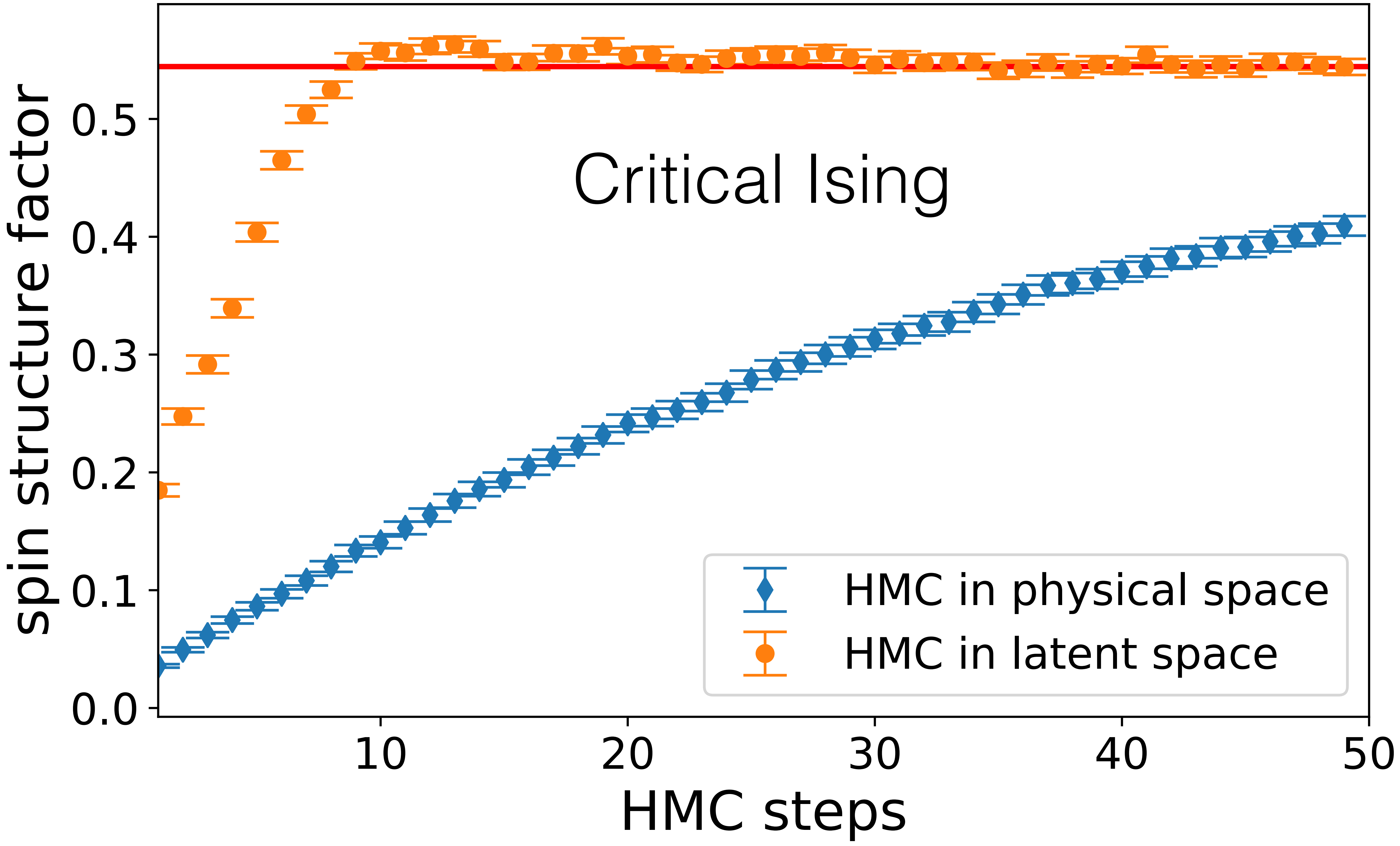
Learnable change-of-variables for
a mutually independent representation

Latent space HMC



$$E(\boldsymbol{x}) = -\ln \pi(\boldsymbol{x})$$

Physical energy function

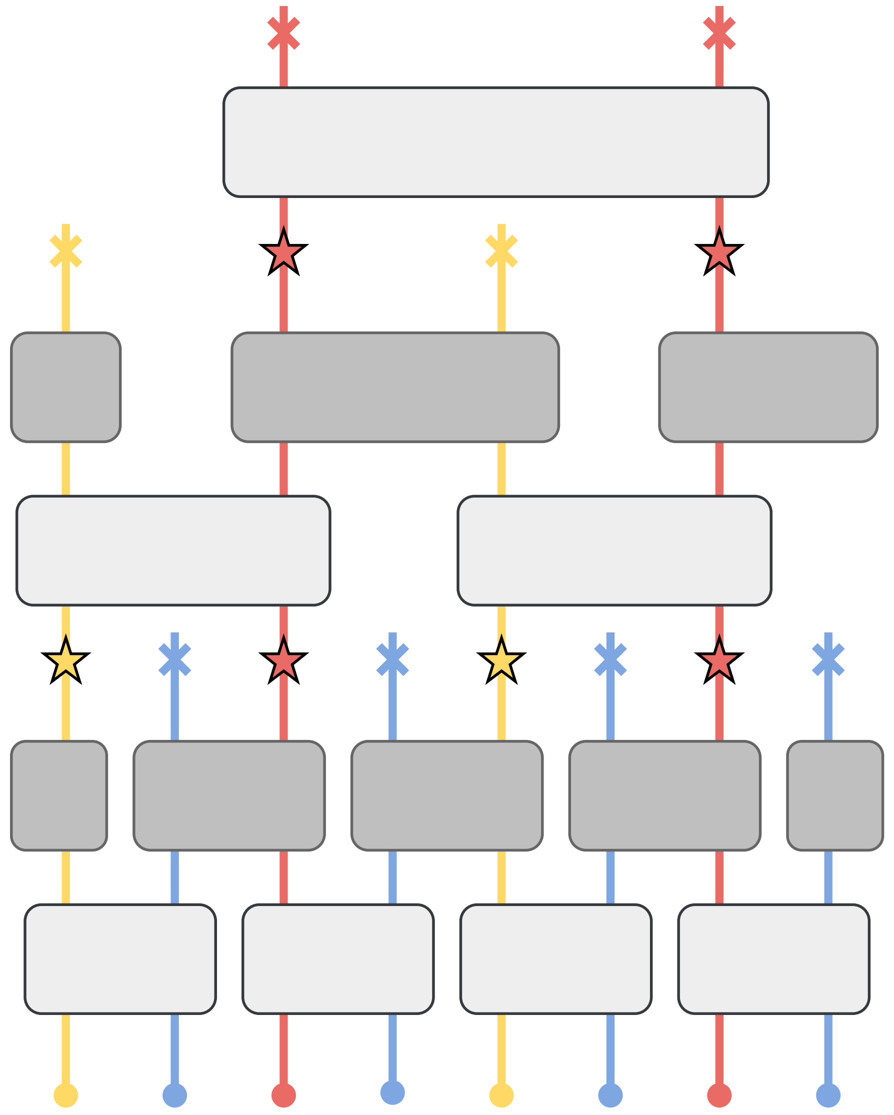


HMC thermalizes faster in the latent space

Latent space HMC

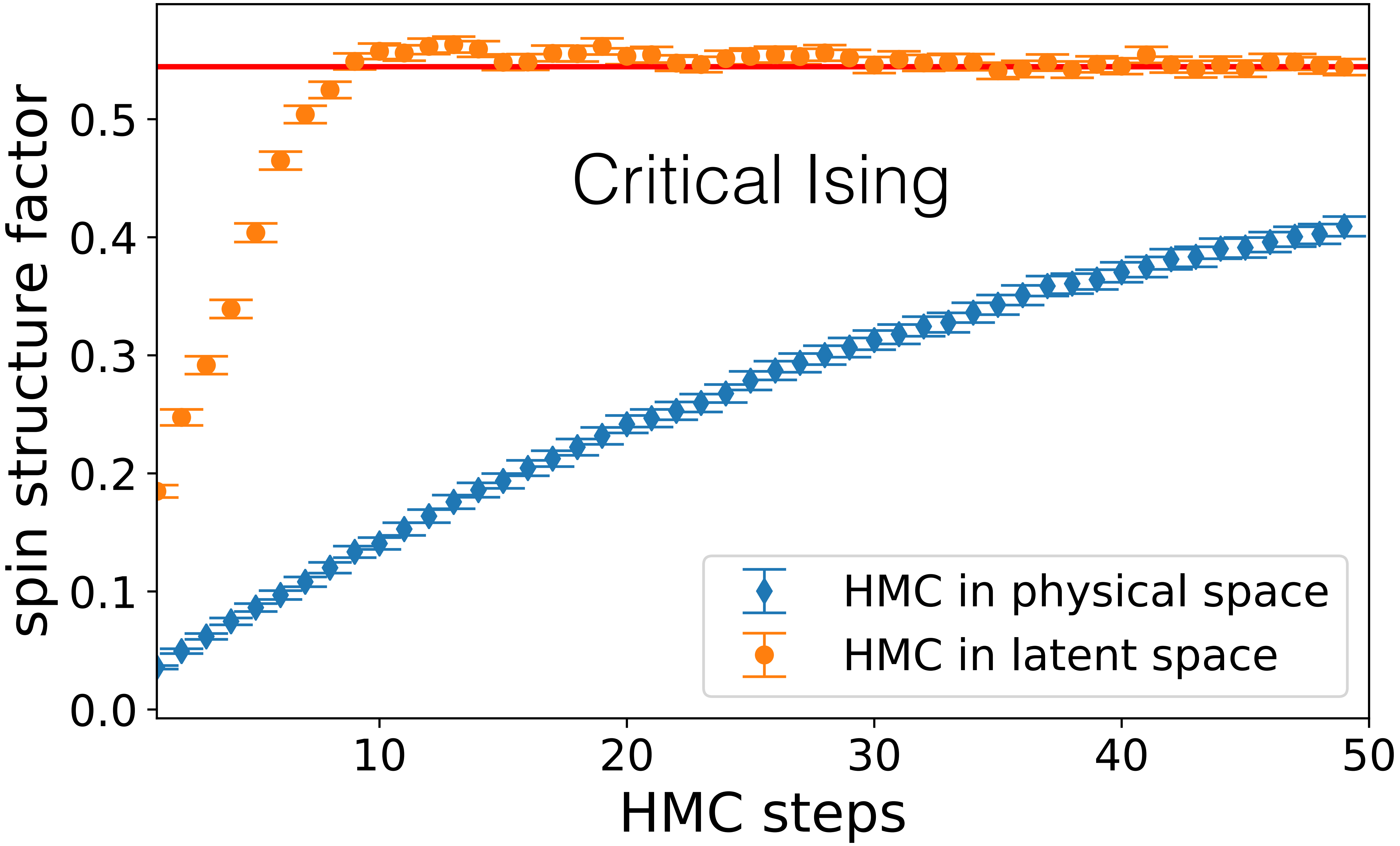
Latent space energy function

$$E(z) = -\ln \pi(g(z)) + \ln q(g(z)) - \ln p(z)$$



$$E(\boldsymbol{x}) = -\ln \pi(\boldsymbol{x})$$

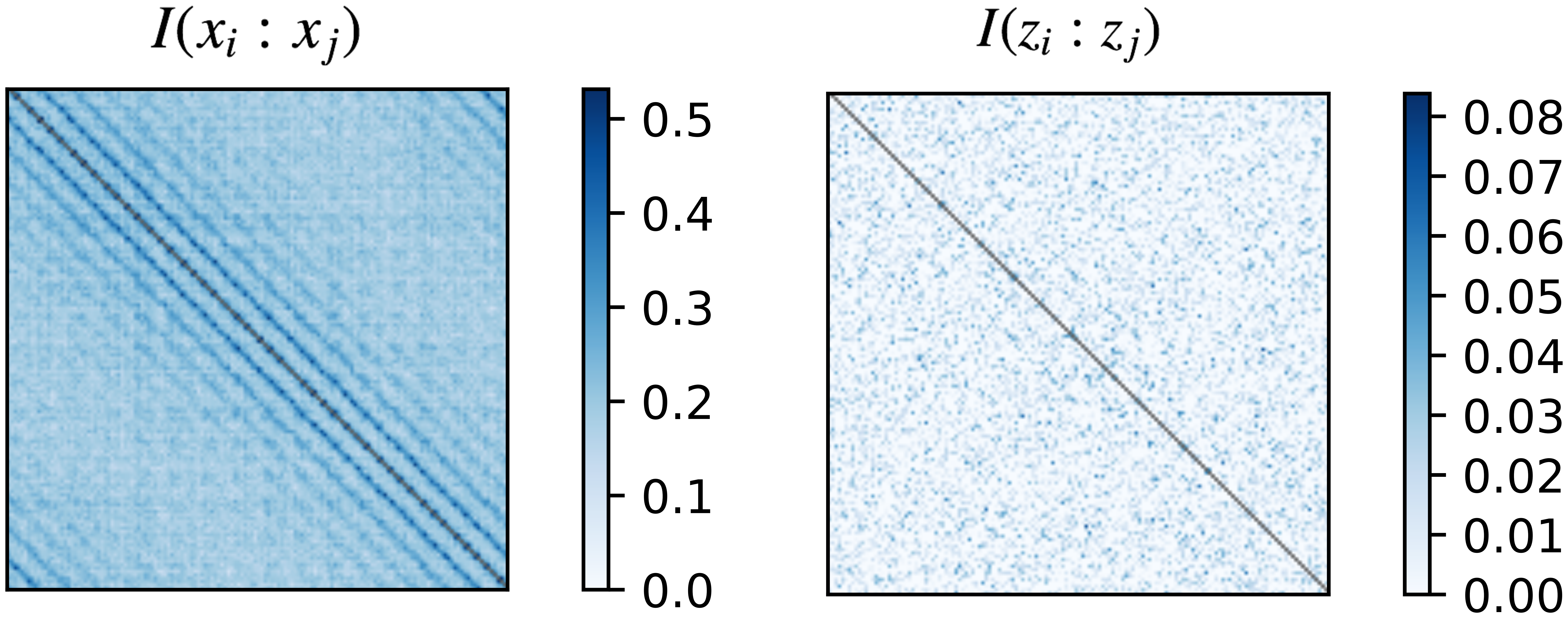
Physical energy function



HMC thermalizes faster in the latent space

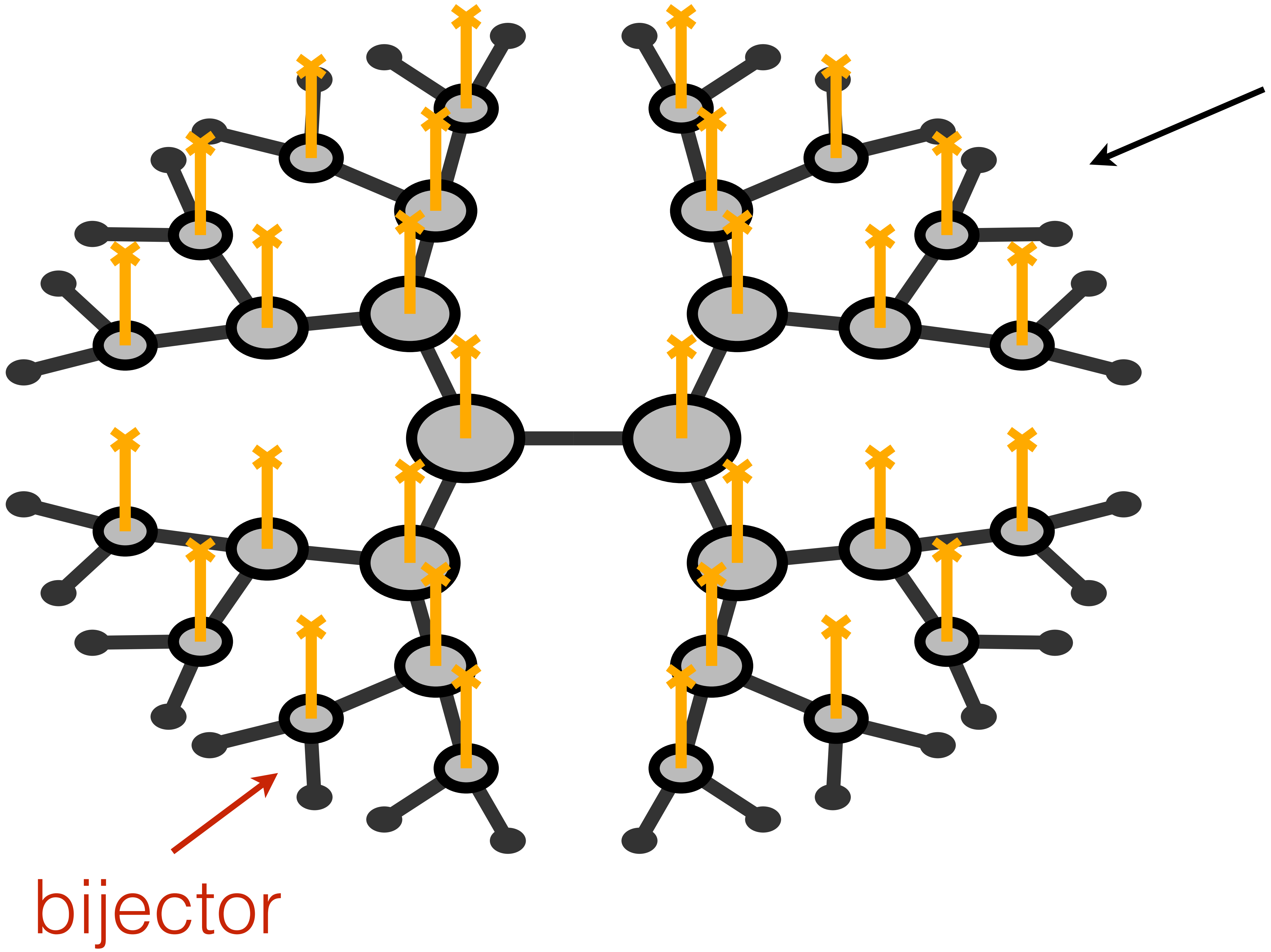
Mutual information

KSG MI estimator
Phys. Rev. E 69, 066138 (2004)



Reduced Mutual Information in the latent space

MI and holographic RG



This is a neural network

Physical variables on the boundary

Latent variables in the bulk

RG flows along the radial direction

Information is preserved by the flow

Qi 1309.6282, You, Qi, Xu 1508.03635
You, Yang, Qi 1709.01223

Normalizing flow implements an invertible RG flow

Mutual information reveals the emergent geometry in the bulk

Remarks on RG

- Conventional RG fixes the transformation and searches for the fixed point. Now, learn the transformation towards the Gaussian fixed point.

- Conventional RG is a semi-group. Here, it is a group builds on bijectors. Coarse-graining is done by the hierarchical network architecture (Wegner 74').

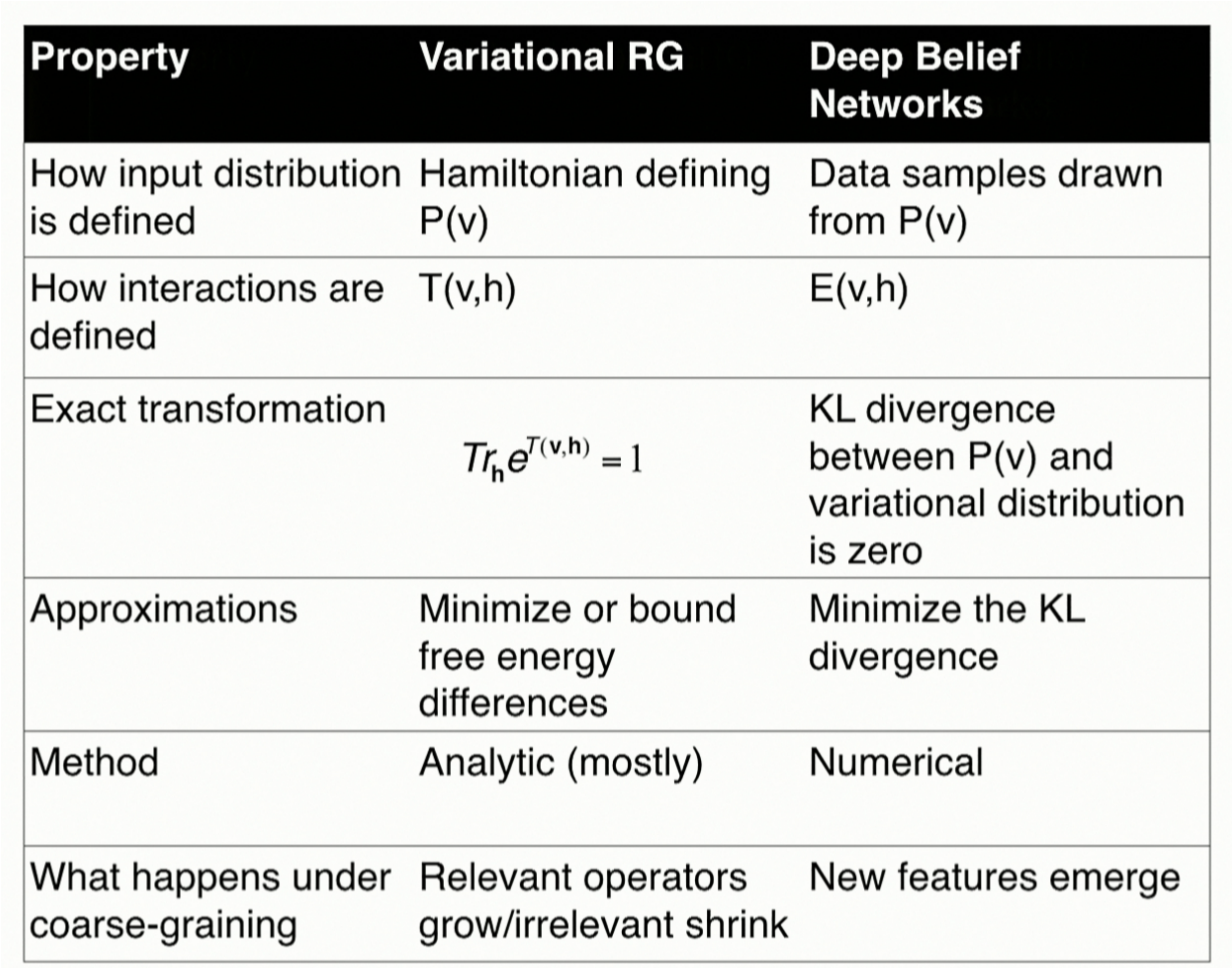
- Changes of variables formulation of RG (Caticha 16')

- Probabilistic (Jona-Lasinio 75') and Information Theory (Apenko 09') perspectives on RG (same is true for neural & tensor networks)

More Remarks

- Learns from bare energy function, instead of training data
- Extends conventional RG with modern DL technique, and with a different goal
- Is a practical computational tool for realistic systems
- Does not seem to be strong for universality, exponents and so on
- Can be regarded as an implementation of the insights of Bény 13’.

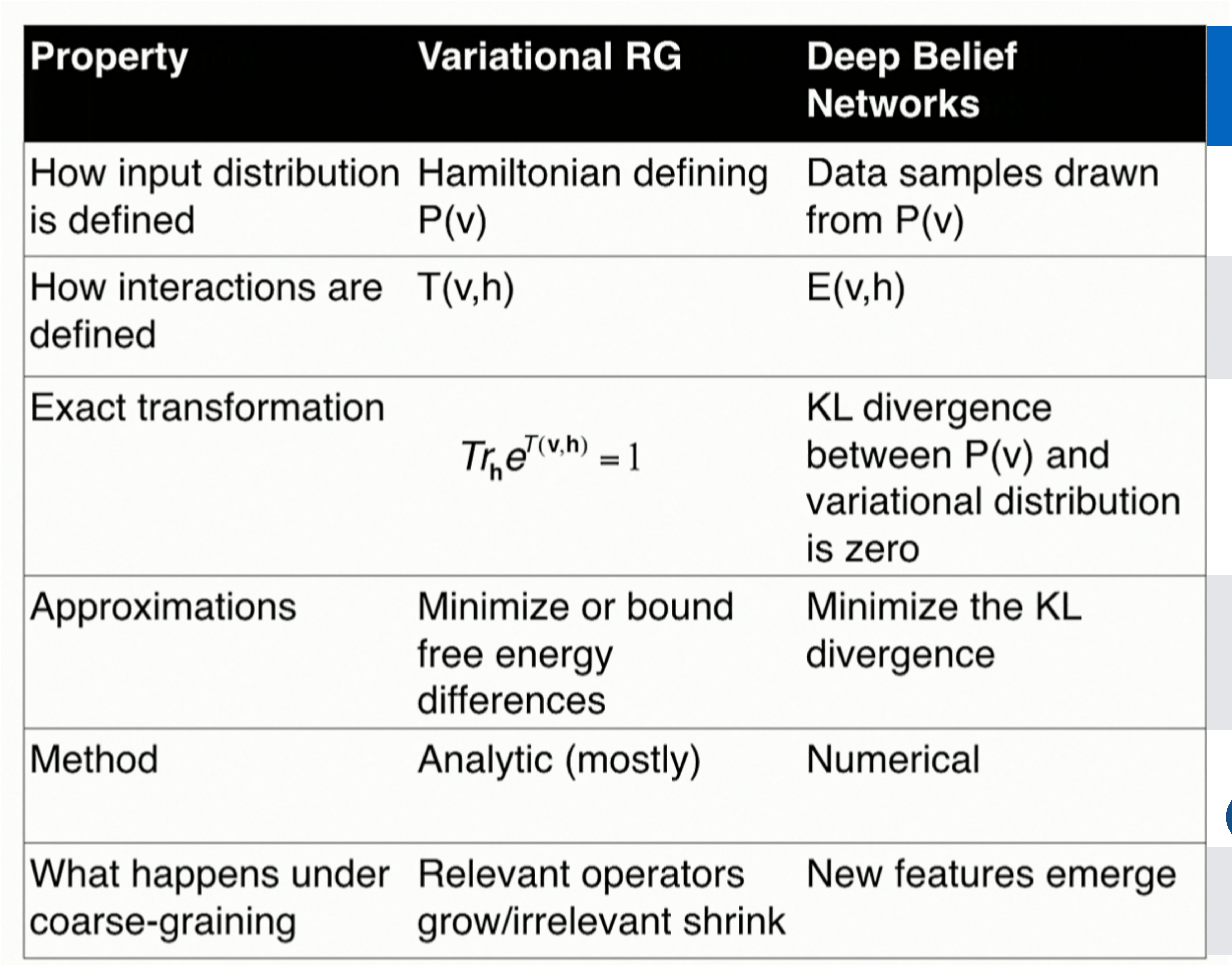
Dictionary: RG vs Deep Learning



Property	Variational RG	Deep Belief Networks
How input distribution is defined	Hamiltonian defining P(v)	Data samples drawn from P(v)
How interactions are defined	T(v,h)	E(v,h)
Exact transformation	$Tr_{\mathbf{h}} e^{T(\mathbf{v},\mathbf{h})} = 1$	KL divergence between P(v) and variational distribution is zero
Approximations	Minimize or bound free energy differences	Minimize the KL divergence
Method	Analytic (mostly)	Numerical
What happens under coarse-graining	Relevant operators grow/irrelevant shrink	New features emerge

Table from Schwab’s talk at PI: http://pirsa.org/displayFlash.php?id=16080006

Dictionary: RG vs Deep Learning



Property	Variational RG	Deep Belief Networks	Normalizing Flow
How input distribution is defined	Hamiltonian defining P(v)	Data samples drawn from P(v)	**Bare energy function**
How interactions are defined	T(v,h)	E(v,h)	**Nonlinear bijectors**
Exact transformation	$Tr_h e^{T(\mathbf{v},\mathbf{h})} = 1$	KL divergence between P(v) and variational distribution is zero	**Reverse KL divergence reaches zero**
Approximations	Minimize or bound free energy differences	Minimize the KL divergence	**Variational minimization of the free energy**
Method	Analytic (mostly)	Numerical	**Numerical (Differentiable Programming)**
What happens under coarse-graining	Relevant operators grow/irrelevant shrink	New features emerge	**Progressly decoupled degrees of freedom**

Table from Schwab's talk at PI: http://pirsa.org/displayFlash.php?id=16080006

Remarks on accelerated MC

1. Cheap surrogate function for Metropolis rejection: Neal 96’ Jun. S Liu 01’

2. Recommender engine for MC updates using generative models: Huang, LW, 1610.02746, Liu, Qi, Meng, Fu, 1610.03137

   Junwei’s talk on Monday
   Kai’s & Nobu’s posters

3. Reinforcement learning the transition kernel: Song et al, 1706.07561, Levy et al 1711.09268, Cusumano-Towner et al 1801.03612

   Ying-Jer’s poster

4. Performs MC in the learned disentangled representation: Wavelet MC, Ismail 03’

   Present approach

Remarks on tensor networks

- What we had is a classical downgrade of MERA Bény 2013

  Probability Density~ Quantum Wavefuntion

  Classical Mutual Information ~ Entanglement Entropy

  “Decorrelator" ~ Disentangler

  Decimator~Isometry

  Bijectivity~Unitary

- RG transformation is done via normalizing flow (composition of bijectors), instead of tensor operations

- Deep Learning machinery provides structural flexibility, modular abstraction, and end-to-end differentiable learning

- TNS gives back to DL an understanding of what are they doing (and hopefully, how to do better)

Remarks on Deep Learning

Old Wisdoms

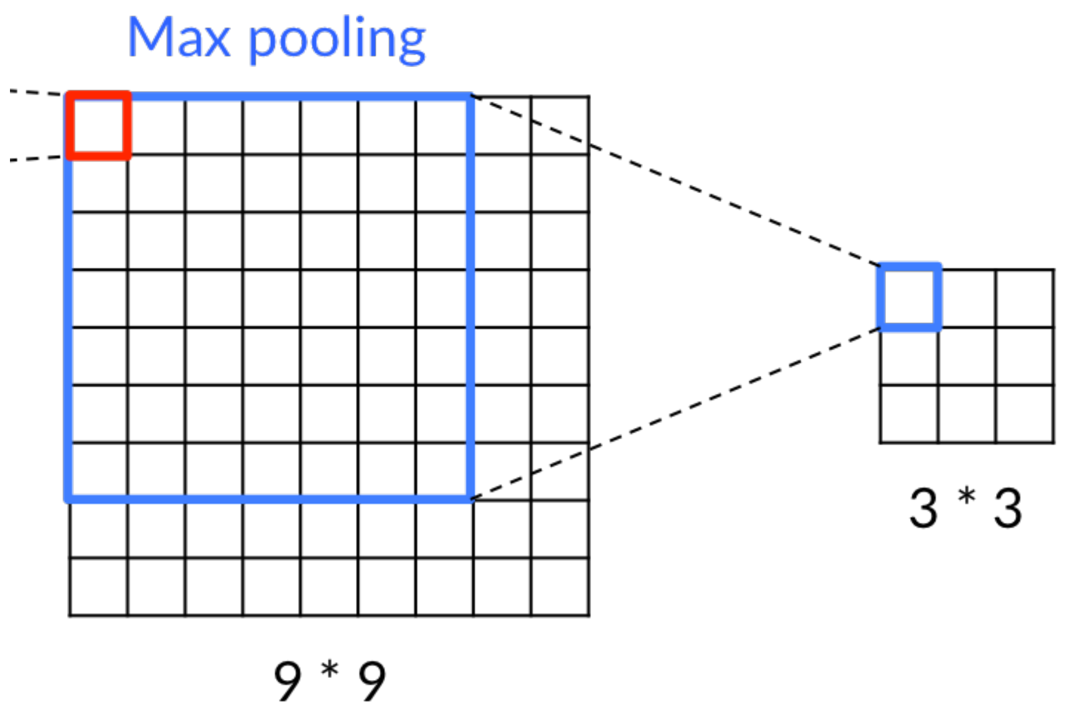
Pooling layer in ConvNets ~ Decimation

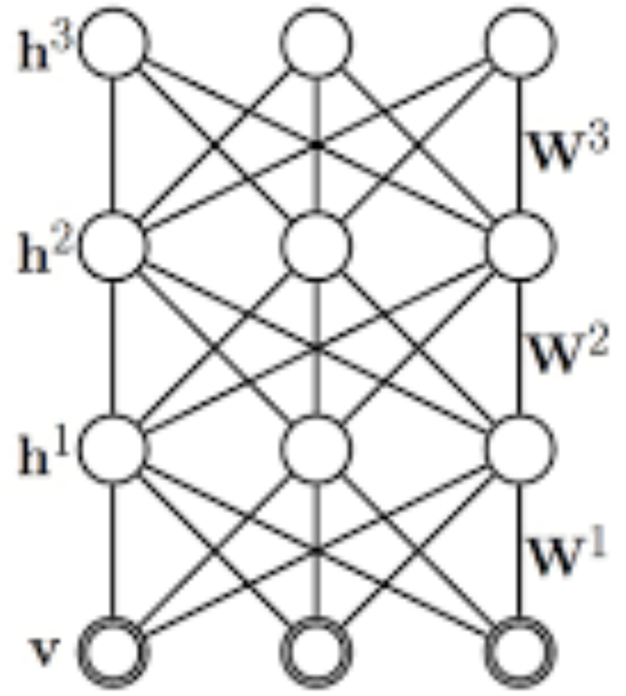
Hidden nodes of deep energy-based model ~ Renormalized Variables

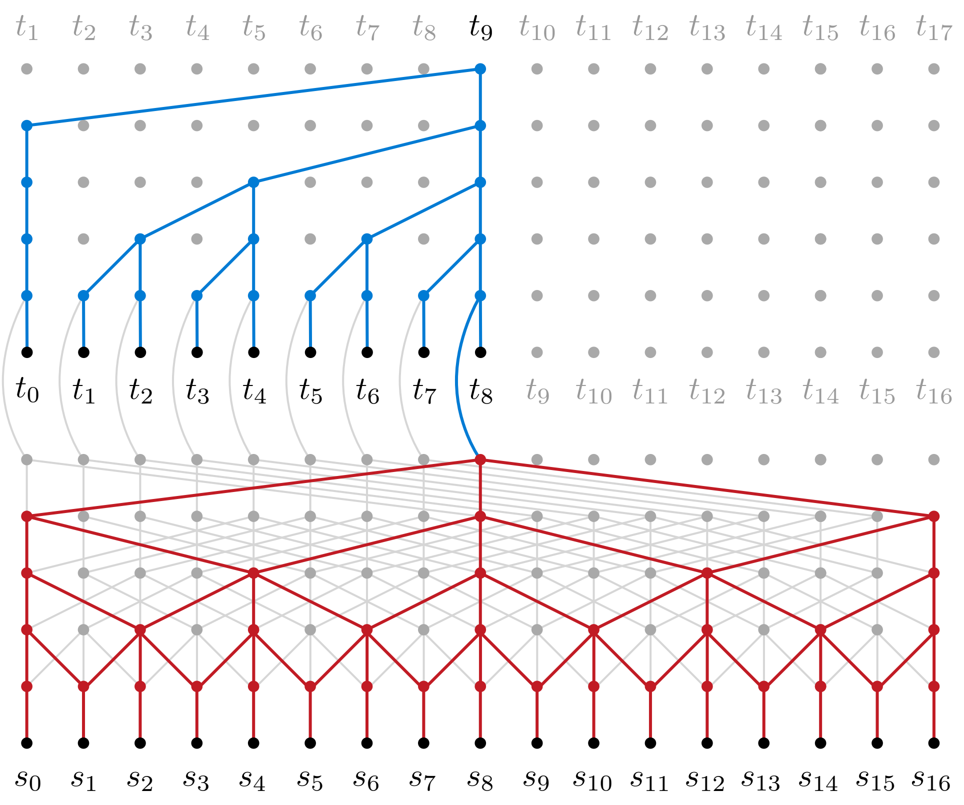
New Insights

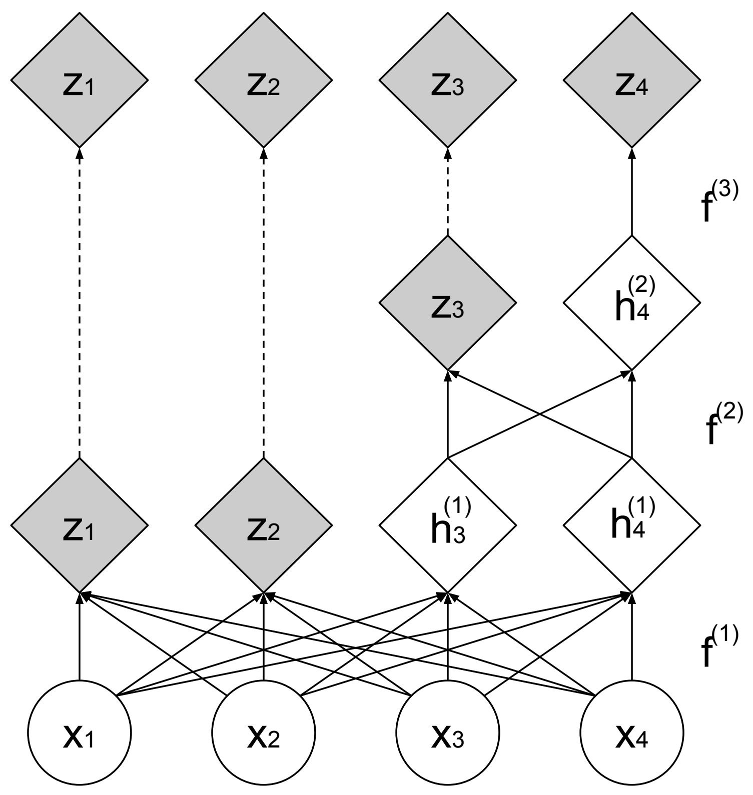
Dilated convolution or Factor out layers = Decimation

Latent variables in the normalizing flow= Renormalized Variables

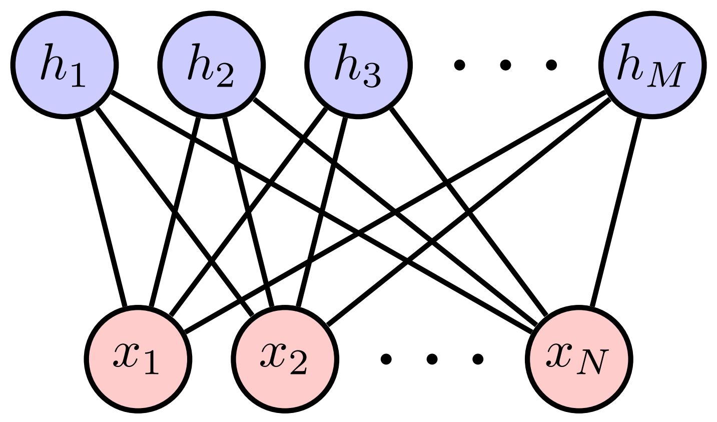


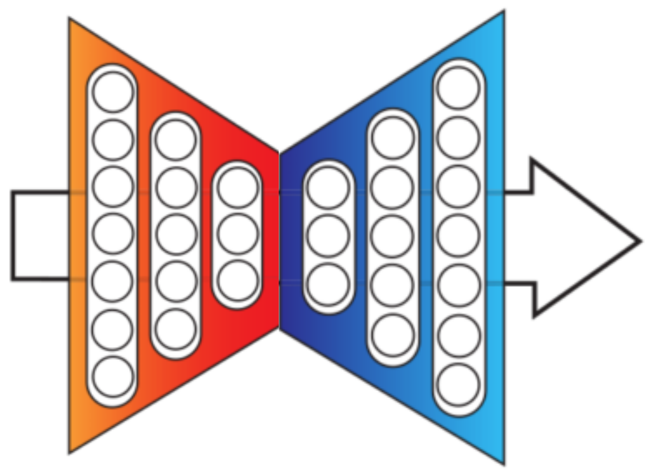


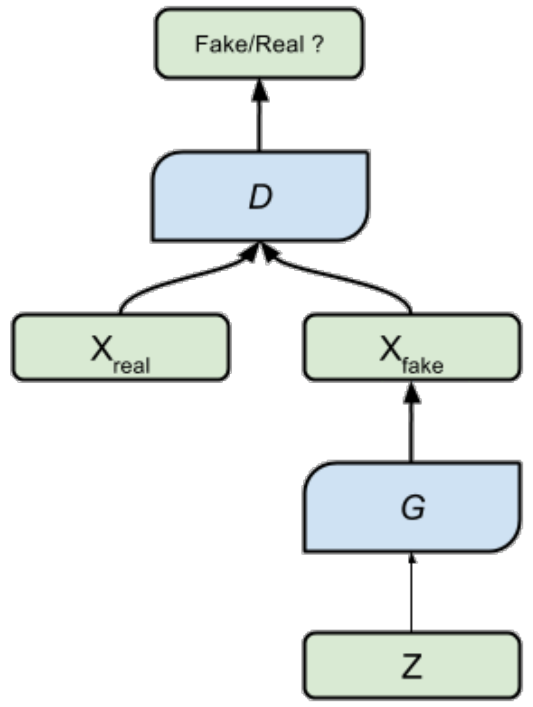

t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16} t_{17}
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13} t_{14} t_{15} t_{16}
s_0 s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15} s_{16}

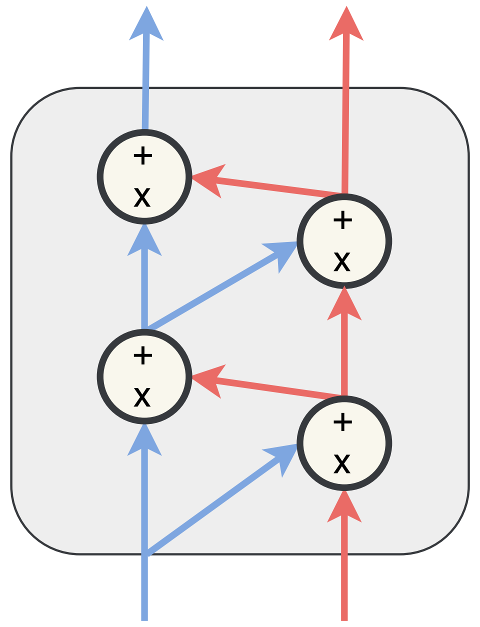


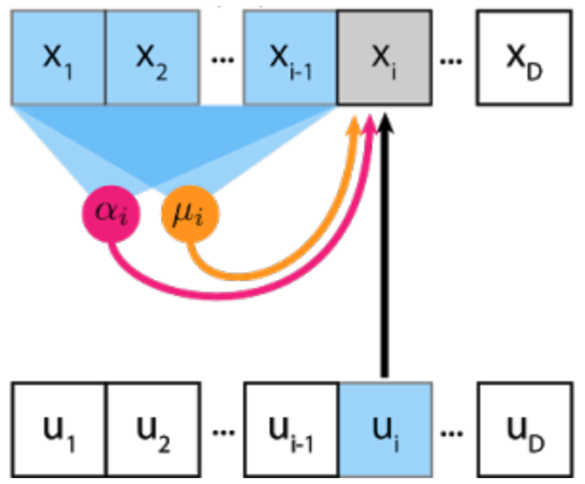
Remarks on Generative Models

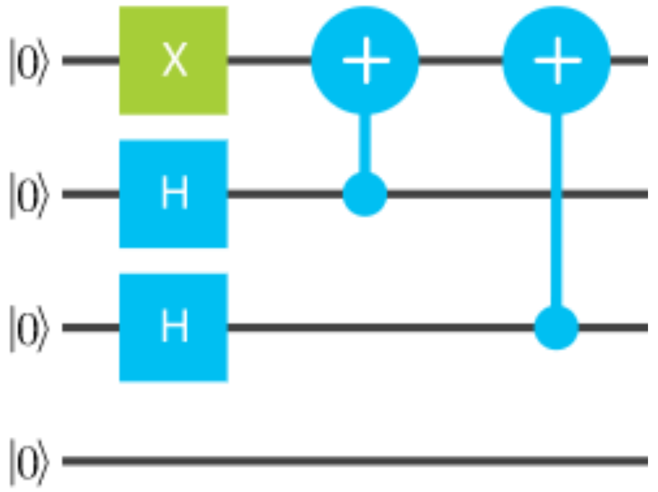












Boltzmann Machines 1980s

Variational Autoendoer 2013

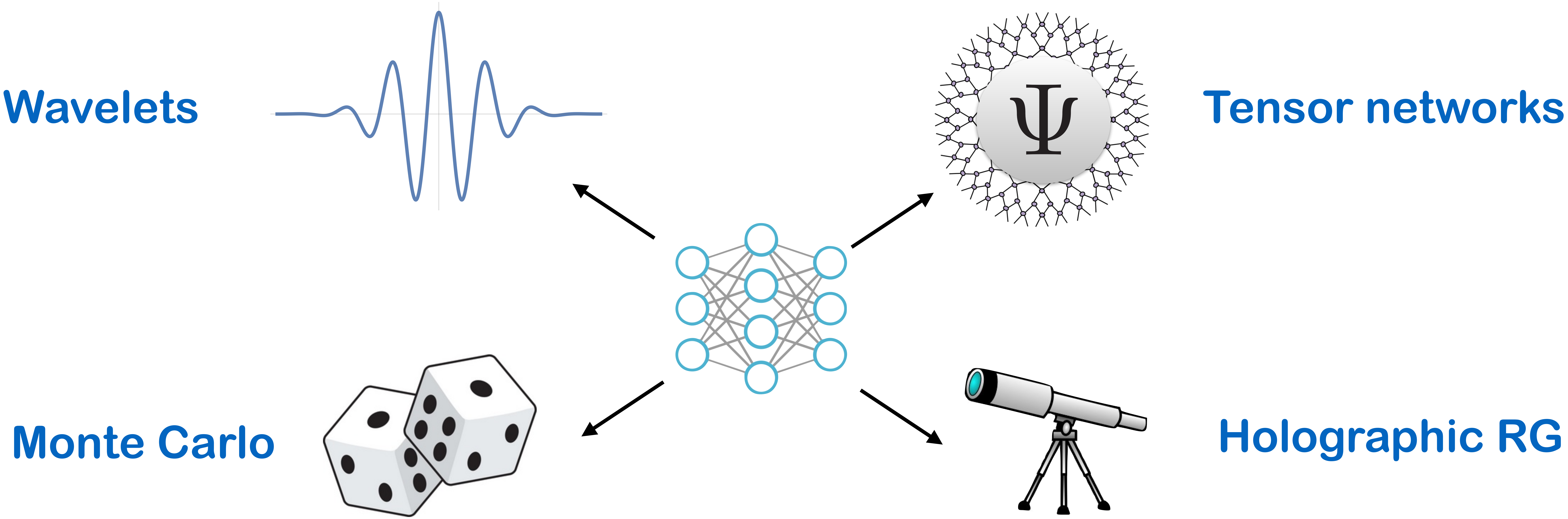
Adversarial Network 2014

Normalizing Flows 2015

Autoregressive Flows 2016

Born Machines 2017

Leverage the power of modern generative models for physics



Wavelets

Tensor networks

Monte Carlo

Holographic RG

Thank You!

Shuo-Hui Li Jin-Guo Liu Pan Zhang Yi-Zhuang You

IOP, CAS ITP, CAS UCSD